A Program Logic for Concurrent Randomized Programs in the Oblivious Adversary Model

Weijie Fan[1], Hongjin Liang[1]✉, Xinyu Feng[1], and Hanru Jiang[2]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, Jiangsu, China
weijiefan@smail.nju.edu.cn, {hongjin, xyfeng}@nju.edu.cn
[2] Beijing Institute of Mathematical Sciences and Applications, Beijing 101408, Beijing, China
hanru@bimsa.cn

Abstract. Concurrent randomized programs in the oblivious adversary model are extremely difficult for modular verification because the interaction between threads is very sensitive to the program structure and the execution steps. We propose a new program logic supporting thread-local verification. With a novel "split" mechanism, one can split the state distribution into smaller partitions, and the reasoning can be done based on each partition independently, which allows us to avoid considering different execution paths of **if**-statements and **while**-loops simultaneously. The logic rules are compositional and are natural extensions of their sequential counterparts. Using our program logic, we verify four typical algorithms in the oblivious adversary model.

1 Introduction

Randomization has become an important and powerful technique in the design of concurrent and distributed algorithms. By introducing probabilistic coin-flip operations, problems like consensus and leader election can be solved efficiently (e.g. [12,2,3]), despite being inherently difficult or even impossible to solve in a non-probabilistic concurrent setting.

To understand the semantics of concurrent randomized programs, one has to take into account the interplay between concurrency and randomization. In particular, one must answer the question: can the result of a coin-flip operation affect the choice of scheduling (i.e. which thread will perform the next operation)? For this, algorithm designers propose a spectrum of *adversary models* specifying the knowledge about the past execution that a scheduler (a.k.a. an adversary) can use for choosing the next thread. Different adversary models assume different knowledge, varying from none to all.

At one end of the spectrum is the *oblivious adversary* (OA) model, where an adversary has no knowledge and must fix the entire schedule prior to the execution. The OA model is a natural abstraction of most real-world scheduling algorithms, including the round-robin scheduling and the priority-based scheduling.

It reflects the scheduling in almost all real general infrastructures such as operating systems or programming languages (e.g. as in golang) where the scheduling does not rely on the specific behaviors of the threads being scheduled.

Designing algorithms for the OA model has gained lots of attention and more than ten algorithms have been proposed over the years (see [4,5] for a comprehensive introduction). As a concrete example, consider Chor et al. [12]'s *conciliator* algorithm. A conciliator is a weak consensus object that guarantees probabilistic agreement, namely that with a high probability the return values of all threads are equal. In Chor et al. [12]'s conciliator algorithm, each thread i executes C_i:

$$C_i \stackrel{\text{def}}{=} (\textbf{while } (s = 0) \textbf{ do } \langle s := i \rangle \oplus_p \langle \textbf{skip} \rangle)\,; y_i := s$$

Here s is a shared variable initialized to 0, y_i is the local variable for thread i that records its return value. The probabilistic choice $\langle s := i \rangle \oplus_p \langle \textbf{skip} \rangle$ says that thread i writes i to s with probability p and does nothing (**skip**) with probability $1 - p$. It repeats until the thread observes $s \neq 0$, then it loads s to y_i. Given n threads running the conciliator code in the OA model, the algorithm ensures the postcondition $\mathbf{Pr}(y_1 = y_2 = \cdots = y_n) \geq (1-p)^{n-1}$, i.e. the probability for the threads to reach a consensus (thus $y_1 = y_2 = \cdots = y_n$) is no less than $(1-p)^{n-1}$.

However, there has been little attention paid to verifying algorithms in the OA model. Existing program logics for verifying concurrent randomized programs [19,17,13] work with only the *strong adversary* (SA) model, which is at the other end of the spectrum of adversary models. A strong adversary has the full knowledge of the past execution, including outcomes of past coin-flips, thread-local states and shared states. Consequently, any algorithm which is correct under SA must still be correct under OA, but not vice versa. For instance, the aforementioned conciliator algorithm is *not* correct in SA and we will explain why in Sec. 6. None of the existing program logics can apply to the conciliator, or more generally, to any algorithms which are correct only with weaker adversaries such as OA.

On the one hand, it is unclear how to *take advantage of* the OA model in the verification. On the other hand, the OA model brings its own verification challenges. As we will see in Sec. 3, the program behaviors in the OA model seem sensitive to the number of execution steps in different program branches, but the verification with program logics must be modular, syntax-directed and insensitive to the number of execution steps.

The good news is, from the existing algorithms designed for the OA model, we observe that the correctness properties of these algorithms usually follow certain common patterns and can be specified by what we call "closed" assertions, which will be introduced in Sec. 3.2. To verify these properties, we do not need to prove they hold over the whole state distribution, which may contain states resulting from the execution of different program branches. Instead, we can prove there exists a partition of the distribution such that the property holds over every part. For *closed* assertions, the validity over every part implies the validity over the whole distribution.

Based on this observation, we propose the first program logic for concurrent randomized programs targeting the OA model. Our work makes the following new contributions:

- We take advantage of the OA model by proposing an abstract small-step operational semantics over state distributions, which allows us to apply classical concurrency reasoning techniques (such as invariants) by interpreting assertions over state distributions.
- We propose a novel proof technique called *split* to support modular reasoning and overcome the problem with branch statements. By splitting a state distribution into several smaller ones, we can reason about the different program branches independently. This leaves us only to prove the postcondition holds over a partition of the final state distribution. Then we can derive it for the whole distribution as long as the postcondition is closed.
- We design a set of logic rules for compositional reasoning about concurrent randomized programs with the split mechanism. Thanks to the split idea, our rules for sequential composition, **if**-statements and **while**-loops are simple and natural extensions of their classical (non-probabilistic) counterparts.
- We prove that our logic ensures partial correctness of concurrent randomized programs where the adversaries are also randomized. Since we focus on closed assertions as postconditions, the verification is independent of the distribution of schedules. The partial correctness verified by the logic holds over arbitrary probabilistic distributions of oblivious adversaries.
- Using our logic, we report the first formal verification of four typical algorithms in the OA model, including the aforementioned conciliator [12], group election (the core phase of Alistarh and Aspnes' randomized test-and-set algorithm [2]), a shared three-sided dice and a multiplayer level-up game.

Outline. Below we first review mathematical preliminaries in Sec. 2. Then we informally explain our key ideas in Sec. 3. We present the language setting including our abstract semantics in Sec. 4. We develop our program logic in Sec. 5, and verify conciliator as a case study in Sec. 6. We discuss related work in Sec. 7. The appendix contains the full formal details, including semantics rules, logic rules and soundness proofs, and examples.

2 Preliminaries

Below we review the background on probability theory and sketch the basic mathematical notations used in our work for describing probabilities, expected values, etc. Readers who are not interested in mathematics can safely skip this section and come back later when the notations are used.

A *sub-distribution* over a set A is defined as a function $\mu : A \to [0, 1]$ such that

- the support $\mathit{supp}(\mu) \stackrel{\text{def}}{=} \{a \in A \mid \mu(a) > 0\}$ is countable; and
- the weight $|\mu| \stackrel{\text{def}}{=} \sum_{a \in A} \mu(a)$ is less than or equal to 1.

If we have $|\mu| = 1$, we say μ is a *distribution* over A. We use $\mathbb{SD}_A$ to denote the set of sub-distributions over A, and $\mathbb{D}_A$ to denote the set of distributions. For $\mu \in \mathbb{SD}_A$, intuitively $\mu(a)$ represents the *probability* of drawing a from μ.

We define the *probability of an event* $E : A \to \text{Prop}$ and the *expected value of a random variable* $V : A \to \mathbb{R}$ as follows, denoted by $\mathbf{Pr}_{a\sim\mu}[E(a)]$ and $\mathbb{E}_{a\sim\mu}[V(a)]$ respectively (where a is a bound variable, just like $\sum_{a\in A} f(a)$). Here Prop represents the set of propositions, and $\mathbb{R}$ is the set of real numbers.

$$\mathbf{Pr}_{a\sim\mu}[E(a)] \stackrel{\text{def}}{=} \sum_{a\in A}\{\mu(a) \mid E(a)\} \qquad \mathbb{E}_{a\sim\mu}[V(a)] \stackrel{\text{def}}{=} \sum_{a\in A} \mu(a)\cdot V(a) \tag{1}$$

For instance, suppose μ is a state distribution, and $\mathbf{q}$ is a state assertion (we write $\sigma \models \mathbf{q}$ if $\mathbf{q}$ holds at the state σ). Then $\mathbf{Pr}_{\sigma\sim\mu}[\sigma \models \mathbf{q}]$ represents the probability that $\mathbf{q}$ is satisfied. If $[\![e]\!]_\sigma$ is the evaluation of the expression e on σ, then $\mathbb{E}_{\sigma\sim\mu}[[\![e]\!]_\sigma]$ represents the expected value of e in μ.

For an event E with non-zero probability in μ (i.e. $\mathbf{Pr}_{a\sim\mu}[E(a)] > 0$), we define the *conditional sub-distribution* $\mu|_E$ as follows:

$$\mu|_E \stackrel{\text{def}}{=} \lambda a. \begin{cases} \frac{\mu(a)}{\mathbf{Pr}_{a\sim\mu}[E(a)]}, & \text{if } E(a) \text{ holds} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Given two sub-distributions $\mu_1, \mu_2 \in \mathbb{SD}_A$ and a probability $p \in [0, 1]$, we define the *mixture sub-distribution* $\mu_1 \oplus_p \mu_2 \in \mathbb{SD}_A$ as follows:

$$\mu_1 \oplus_p \mu_2 \stackrel{\text{def}}{=} \lambda a.\ p\cdot\mu_1(a) + (1-p)\cdot\mu_2(a) \tag{3}$$

Given two sub-distributions $\mu_1 \in \mathbb{SD}_A$ and $\mu_2 \in \mathbb{SD}_B$, we define the *product sub-distribution* $\mu_1 \otimes \mu_2 \in \mathbb{SD}_{A\times B}$ as follows:

$$\mu_1 \otimes \mu_2 \stackrel{\text{def}}{=} \lambda(a, b).\ \mu_1(a)\cdot\mu_2(b) \tag{4}$$

In Sec. 4.2, we will use the product $\otimes$ to compute the initial distribution of program configurations, from the initial program $\mathbb{C}$ and an initial state distribution. When $\mathbb{C}$'s execution ends, we will extract the final state distribution from the final distribution of program configurations by projection. Specifically, given $\mu \in \mathbb{SD}_{A\times B}$, the *projection* of μ with the sets A and B is defined as:

$$\mu^{(A)} \stackrel{\text{def}}{=} \lambda a'.\mathbf{Pr}_{(a,b)\sim\mu}[a = a'] \qquad \mu^{(B)} \stackrel{\text{def}}{=} \lambda b'.\mathbf{Pr}_{(a,b)\sim\mu}[b = b'] \tag{5}$$

For *almost surely terminating* programs (i.e. programs which have infinite executions with zero probability and terminate with probability 1), we define the "final" state distribution as the limit of an infinite sequence of state distributions. In general, we define the limit of a convergent sequence of sub-distributions in Def. 6.

Definition 6 (convergent sequence of sub-distributions). Let A be a set, $\vec{\mu}$ be an infinite sequence of sub-distributions over A. We say $\vec{\mu}$ *converges to a sub-distribution* μ, represented as $\lim\vec{\mu} = \mu$, if and only if $\lim_{n\to\infty}\sum_{a\in A}|\vec{\mu}[n](a) - \mu(a)| = 0$ (where $\vec{\mu}[n]$ means the n-th element of the sequence $\vec{\mu}$). We say $\vec{\mu}$ diverges and $\lim\vec{\mu}$ is undefined if $\vec{\mu}$ does not converge to any μ.

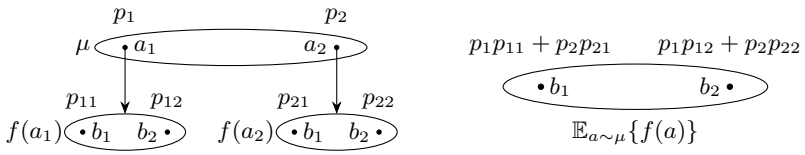


Fig. 1: Expected sub-distribution

Definition 7 (expected sub-distribution). Let $\mu \in \mathbb{SD}_A$ and $f : A \to \mathbb{SD}_B$. The *expected sub-distribution* $\mathbb{E}_{a\sim\mu}\{f(a)\} \in \mathbb{SD}_B$ is defined as

$$\mathbb{E}_{a\sim\mu}\{f(a)\} \stackrel{\text{def}}{=} \lambda b. \sum_{a\in A} \mu(a) \cdot f(a)(b)$$

Definition 7 computes the sub-distributions' expectation. As illustrated in Fig. 1, the function f transforms each element a_i in the support of μ to a sub-distribution $f(a_i)$, and then the expected sub-distribution (see the right side of the figure) is the mixture of all $f(a_i)$.

Also, from a sub-distributions' sub-distribution $\mu \in \mathbb{SD}_{\mathbb{SD}_A}$, we can compute the *flattened sub-distribution* $\overline{\mu} \in \mathbb{SD}_A$ as the mixture of all the sub-distributions in the support of μ:

$$\overline{\mu} \stackrel{\text{def}}{=} \lambda a. \sum_{\nu\in supp(\mu)} \mu(\nu) \cdot \nu(a)\,. \tag{8}$$

3 Informal Development

Below we start with reasoning about sequential randomized programs (Sec. 3.1). For concurrent randomized programs, we introduce the oblivious adversary (OA) model and define the correctness of programs with randomized schedules (Sec. 3.2). Then we show how to do thread-local reasoning by taking advantage of OA (Sec. 3.3). To address the challenges posed by branch statements (Sec. 3.4), we propose the split mechanism (Sec. 3.5).

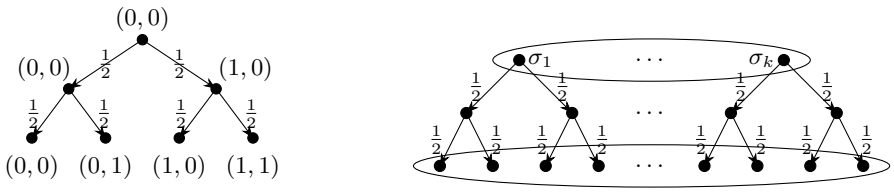
3.1 Sequential Randomized Programs and Their Correctness

Randomized programs can be viewed as programs in a classical (non-probabilistic) programming language (e.g. WHILE) extended with probabilistic choice statements $\langle C_1\rangle \oplus_p \langle C_2\rangle$. It makes a random choice to execute $\langle C_1\rangle$ or $\langle C_2\rangle$, with probability p and $1-p$, respectively. Here we use $\langle C\rangle$ to represent an *atomic* statement that executes C in one step (see the formal semantics in Sec. 4.1).

The execution of a *sequential* randomized program starting from a particular initial state forms a tree. For instance, Fig. 2a shows the execution tree for

$$\mathit{Coins} \stackrel{\text{def}}{=} \langle x := 0\rangle \oplus_{\frac{1}{2}} \langle x := 1\rangle;\ \langle y := 0\rangle \oplus_{\frac{1}{2}} \langle y := 1\rangle;$$

starting from the initial state where x and y are both 0. Each branching in the tree corresponds to a probabilistic choice. If we consider all possible initial states, the execution becomes a forest (where each node represents a program state σ), as shown in Fig. 2b.



(a) (b)

Fig. 2: Execution of a sequential program. In (a), a pair at a node specifies x and y's values in the state.

Correctness. Although the execution model based on the view of state transitions is similar to the model of classical sequential programs, the properties of randomized programs can be significantly different. For the program *Coins*, one may want to derive properties like "the probability that x equals y at the end of the program is 0.5". Unlike a postcondition in Hoare-style logics for classical sequential programs, which is expected to hold over *every* leaf node of the forest, the above property describes *the collection of all the leaf nodes* as a whole, i.e. the state distribution at the end of the program.

Therefore, in the Hoare-style specification $\{P\}C\{Q\}$ for randomized programs, P and Q are assertions over distributions of initial states and final states, respectively. For the example *Coins*, we can specify the aforementioned property as $\{\mathbf{true}\}\mathit{Coins}\{\mathbf{Pr}(x = y) = 0.5\}$ or $\{\mathbf{true}\}\mathit{Coins}\{\lceil x = y\rceil \oplus_{0.5} \lceil x \neq y\rceil\}$. Here $\lceil\mathbf{p}\rceil$ lifts the state assertion $\mathbf{p}$ to an assertion over *state distributions* μ, requiring that $\mathbf{p}$ holds at all states in $\mathit{supp}(\mu)$. The assertion $P \oplus_p Q$ holds at μ, if μ is a *mixture* of two distributions μ_0 and μ_1, which are associated with probabilities p and $1{-}p$, and satisfy P and Q respectively. We can give the following Hoare-logic rule to probabilistic choices:

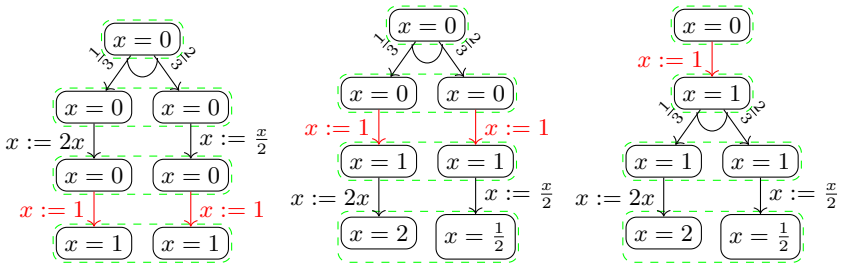
$$\frac{\vdash_{\text{SQ}} \{P\}C_1\{Q_1\} \qquad \vdash_{\text{SQ}} \{P\}C_2\{Q_2\}}{\vdash_{\text{SQ}} \{P\}\langle C_1\rangle \oplus_p \langle C_2\rangle\{Q_1 \oplus_p Q_2\}} \ (\text{SQ-PCH})$$

In this view, a program C transforms a state distribution μ satisfying P to another μ' satisfying Q (an alternative view is expectation-based, where P and Q are expectations [16,8]). The resulting logic rules (e.g. [6]) are almost the same as the classical (non-probabilistic) ones — we just need to lift the assertions from predicates over states to predicates over state distributions.

3.2 Concurrent Randomized Programs and the OA Model

A concurrent randomized program $C_1 \parallel \cdots \parallel C_n$ (denoted by $\mathbb{C}$) has two sources of nondeterminism: the probabilistic choices (in each thread C_i) and the scheduling. Its correctness usually assumes a certain class of scheduling, specified by an *adversary model*.

The *oblivious* adversary (OA) model considered in this paper requires that the scheduling must be determined prior to the execution, regardless of the outcomes of a thread's local coin-flip operations. For example, Fig. 3 shows all



(a) (b) (c)

Fig. 3: Execution trees in OA model for $\mathbb{C}_x \stackrel{\text{def}}{=} (\langle x := 2x \rangle \oplus_{\frac{1}{3}} \langle x := \frac{x}{2} \rangle \parallel x := 1)$

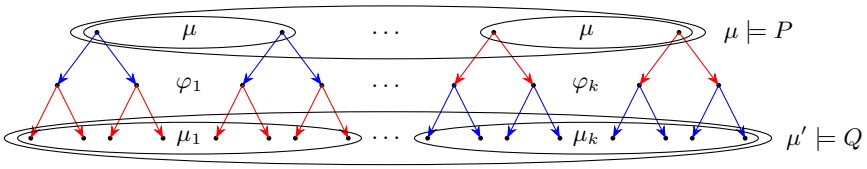


Fig. 4: Illustration of $\models \{P\}\mathbb{C}\{Q\}$

the possible executions in the OA model for a simple program $\mathbb{C}_x$ consisting of two threads: $\langle x := 2x \rangle \oplus_{\frac{1}{3}} \langle x := \frac{x}{2} \rangle \parallel x := 1$. In the concurrent setting, the probabilistic choice $\langle C_1 \rangle \oplus_p \langle C_2 \rangle$ is executed in *two steps*: it first flips a coin, getting heads with probability p and tails with probability $1-p$, and then executes either the atomic statement $\langle C_1 \rangle$ for heads, or $\langle C_2 \rangle$ for tails.

Therefore, in OA, there are only three possible schedules for $\mathbb{C}_x$: $t_1\ t_1\ t_2$ (Fig. 3a); $t_1\ t_2\ t_1$ (Fig. 3b); and $t_2\ t_1\ t_1$ (Fig. 3c). In the figure, state transitions by different threads are in different colors (in black for t_1, and in red for t_2). We can see that, by fixing a specific OA schedule, the transitions at the same layer of an execution tree must be made by the *same* thread.

In contrast, the *strong* adversary (SA) model allows arbitrary scheduling. An SA scheduler has the full knowledge of machine states, especially including the outcomes of coin-flip operations, and can rely on that knowledge to schedule threads. For the example $\mathbb{C}_x$, in addition to the three schedules in Fig. 3, the SA model also allows two additional schedules, where t_1 and t_2 are scheduled in different orders for different outcomes of the coin flip. As such, the transitions at the same layer of an execution tree could be made by *different* threads.

This example also demonstrates that, thanks to the restriction of the scheduling, one can derive stronger properties of programs in the OA model that do not hold in the SA model. As shown in Fig. 3, in the OA model, the expected value of x at the end of execution is 1, which is not true considering the two more schedules in the SA model.

Correctness and closed assertions. What is the meaning of the Hoare triple $\{P\}\mathbb{C}\{Q\}$ now? Figure 4 shows the execution of a concurrent program, where μ

is the distribution of the initial states. We use $\mu \models P$ to denote that μ satisfies P, which will be formally defined in Sec. 5.1. The execution under each (OA) schedule φ_i corresponds to a forest, as in the case for sequential programs. Edges of different colors represent execution steps from different threads. The execution under all schedules forms *a set of forests*. It is obvious that P specifies μ, but what about Q?

Here we have two choices. We can either view the schedules as being *non-deterministic*, or as being *probabilistic*. For the former, we require that Q holds over every μ_i (the leaf node distribution of the forest generated with the schedule φ_i). However, this result is not strong enough — if we sample the execution of $\mathbb{C}$ and observe the final results, the sampled executions may not be generated with the same schedule, that is, the final states we observe may come from different μ_i. So it is more natural to take the latter (probabilistic) view of schedule and consider the mixture distribution μ' of $\mu_1, \ldots, \mu_k, \ldots$, where the weight of each μ_i is the probability of the schedule φ_i. Since we do not know the distribution of schedules in advance, Q needs to hold for all schedule distributions, that is, Q holds over μ' obtained by taking an *arbitrary* probability distribution for $\mu_1, \ldots, \mu_k, \ldots$

We use $\models_{\text{ND}} \{P\}\mathbb{C}\{Q\}$ to represent the semantics of the Hoare triple under the *non-deterministic* view, and $\models_{\text{PR}} \{P\}\mathbb{C}\{Q\}$ for the *probabilistic* view. It is easy to prove the latter implies the former. The reverse does not hold in general, but it holds if Q is "closed". Here $\mathbf{closed}(Q)$ requires that the mixture of any (potentially countably infinite) number of distributions satisfies Q if each of these distributions satisfies Q. (We will formally define $\mathbf{closed}(Q)$ in Sec. 5.1.) As a result, for a closed postcondition, we can reduce the proof of $\models_{\text{PR}} \{P\}\mathbb{C}\{Q\}$ to the proof of $\models_{\text{ND}} \{P\}\mathbb{C}\{Q\}$.

As far as we know, most concurrent randomized algorithms have closed post-conditions. As examples of closed assertions, $\lceil b \rceil$, $\mathbf{Pr}(b) = 0.5$ and $\mathbb{E}(x) = 1 \wedge \lceil x \geq 0 \rceil$ are all closed. So, for the earlier example $\mathbb{C}_x$, it suffices to prove that the leaf distribution of each execution tree in Fig. 3 satisfies $\mathbb{E}(x) = 1 \wedge \lceil x \geq 0 \rceil$.

We give the formal definition of $\models_{\text{ND}} \{P\}\mathbb{C}\{Q\}$ in Sec. 4.1. We show the formal definition of $\models_{\text{PR}} \{P\}\mathbb{C}\{Q\}$ and prove that they are equivalent when Q is closed in Appendix B. In this paper we focus on closed Q's only and omit the subscript ND/PR henceforth. Note that $\mathbf{closed}(Q)$ is *not* an overly strong requirement for practical programs, because it is needed only for the postcondition Q of the *whole program* $\mathbb{C}$. The postconditions for individual statements and threads do not need to be closed.

3.3 Thread-Local Reasoning in OA

The question is, *how to take advantage of the OA model and verify the stronger correctness guarantee of a program by thread-local reasoning, i.e. verifying one thread at a time.*

A natural thought is to extend the sequential reasoning in Sec. 3.1 to concurrency. To this end, we hope to view the execution of a concurrent program as transitions over state distributions, as we do for sequential reasoning. However,

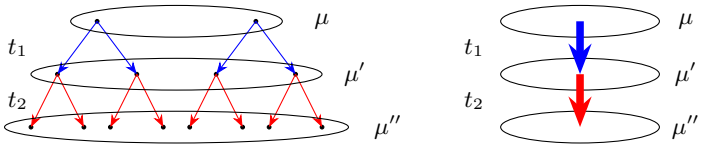


Fig. 5: Concrete *vs.* Abstract Operational Semantics in OA

unlike sequential semantics that are usually *big-step* (see e.g. [6,18]) and care about only the initial and final state distributions, the transitions in a concurrent setting need to be small-step, to reflect the interleaving between threads.

One might also consider to migrate the existing approaches for the SA model to the OA setting. However, the interleaving pattern between threads in the OA model is very different from that in the SA model. The SA model allows that different threads may be scheduled for different outcomes of a probabilistic choice operation, while the OA model does not allow it. As a result, program logics for SA (e.g. [17,13]) adopt *weak* assumptions on the environment behaviors in the thread view: for different states in the support of the current state distribution, different environment threads may interrupt and take very different steps. Therefore, they model the environment behaviors as transitions from states to state distributions (e.g. [17]) or transitions from states to states (e.g. [13]).

However, this idea may not be as useful in the OA setting as in the SA setting (thought it is still sound). Algorithms in the OA model usually rely on the assumption that the scheduling cannot depend on the results of probabilistic choices, so the weak assumption that different states may be interrrupted by different environment threads is too weak in the OA setting, and it is not obvious how to forbid the impossible interleavings in the OA model if we still model the environment behaviors as transitions from states to state distributions or transitions from states to states.

To address this problem, we exploit the stronger assumption on the environment behaviors: for different states in the support of the current distribution, it must be the same environment thread that interrupts and take steps. Therefore, we propose an abstract operational semantics and layer-based reasoning.

Abstract operational semantics. In the OA model, we observe that, for all the states at the same layer of the execution forest (i.e. nodes of the same depths, as shown in Fig. 5a), it is always the same thread picked to execute the next step, since the schedule is predetermined. That is, the edges with the same depths are always of the same color, represents a step from the same thread. Naturally, we can view the states of the same layer as a whole, forming a state distribution. If we also view the edges between two layers as a whole, then Fig. 5a is abstracted to Fig. 5b. This gives us an *abstract operational semantics* with small-step transitions over state distributions. The execution looks like an interleaving execution of a classical (non-probabilistic) concurrent program.

Consequently, we can apply classical concurrency reasoning techniques (e.g. invariants) to reason about executions in our abstract semantics. Our abstraction is sound in that the Hoare-triple $\{P\}\mathbb{C}\{Q\}$ valid in our abstract semantics also holds with the concrete semantics.

Invariants. To do thread-local reasoning, one needs to specify the interference between the current thread and its environment (i.e. the other threads), which can be modeled by an invariant I. For classical concurrent programs, I is a state assertion that needs to hold at all times. The current thread can assume that I holds before each of its steps, but it must also ensure that I still holds after each step. For a randomized program, we define I over state distributions. It holds at all the μ's in executions in our abstract semantics (e.g. μ, μ' and μ'' in Fig. 5b). Since every such μ corresponds to a layer in the concrete semantics, we call I a *layer invariant* and the reasoning layer-based.

In addition to layer invariants I, our logic also uses *non-probabilistic rely and guarantee conditions* R and G, to simplify the formulation of I in proofs of programs. By "non-probabilistic", we mean that R and G specify state transitions in the concrete semantics (but do not specify the probability of the transitions). Their treatment is the same as in classical rely-guarantee reasoning [14].

Unfortunately, we need to address one more challenge to make this nice abstraction work. To define the abstract operational semantics, we view all the edges (program steps) at the same layer in Fig. 5a as a whole to get Fig. 5b. However, although these edges are from the same thread, they may still correspond to the execution of *different code*, due to the branch statements in the thread. Below we explain the challenges and our solution in detail.

3.4 Problems with Branch Statements

A program may contain branch statements such as **if**-statements and **while**-loops, which condition upon random variables (i.e. variables whose values are probabilistic). Different branches may take different numbers of steps to execute, making it difficult to do layer-based reasoning.

For instance, we consider the program $C \parallel c_4$, where:

$$C \stackrel{\text{def}}{=} (\textbf{if } (x = 0) \textbf{ then } (c_{11}; c_{12}) \textbf{ else } c_{21}); c_3;$$

Here each $c_\square$ stands for an atomic command. Assume the initial values of x are assigned in a probabilistic choice, which is either 0 or 1. Figure 6 shows a possible execution, where we need to consider the two possibilities corresponding to the two initial values of x. Note we allow the right branch to execute **skip** when it reaches the end while the left branch executes c_3.

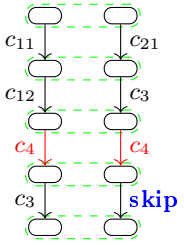


Fig. 6

Thread t_1 switches to t_2 after executing two steps (we omit the step evaluating the boolean condition). The layer-based reasoning asks us to find some invariant and prove that it holds over the distribution of every layer (i.e. every

green dashed box). This forces us to consider the simultaneous execution of c_{11} and c_{21} in the **then**-branch and the **else**-branch. Even worse, since the two branches have different lengths, we have to consider the simultaneous execution of c_{12} and c_3. This looks particularly unreasonable if we consider the fact that c_3 actually sequentially follows c_{12} in the program structure! This makes it almost impossible to design structural and compositional Hoare-style logic rules. The problem is exacerbated by **while**-loops, where the number of rounds of loops may rely on random variables.

Note that this problem does not show up in the deterministic setting where there is no randomization and we prove properties of individual states. In the execution of **if**-statements, a state either enters the **then**-branch or enters the **else**-branch, but not both. So we only need to verify the two cases respectively.

We also do not have to worry about the problem with branch statements in the sequential probabilistic setting. Since there is no interleaving, we can reason about probabilistic properties in a "big-step" flavor where we only consider the initial state distribution and the final one. To reason about the branch statement, we can reason about the different branches (on the corresponding sub-distributions) separately and then do a mixture at the join point, as shown by the (COND) rule in Barthe et al. [6]'s sequential logic:

$$\frac{\{P_1 \wedge \lceil b \rceil\} C_1 \{Q_1\} \qquad \{P_2 \wedge \lceil \neg b \rceil\} C_2 \{Q_2\}}{\{(P_1 \wedge \lceil b \rceil) \oplus (P_2 \wedge \lceil \neg b \rceil)\} \mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2 \{Q_1 \oplus Q_2\}}\ (\textsc{cond})$$

The (COND) rule in [6] is sound for sequential programs, but not for the concurrent OA setting. If C_1 and C_2 have different number of steps, then $Q_1 \oplus Q_2$ specifies a state distribution where states are not at the same "layer", which will make it difficult to reason about subsequent statements.

Below we use an interesting example to further demonstrate the problem and then introduce our solution.

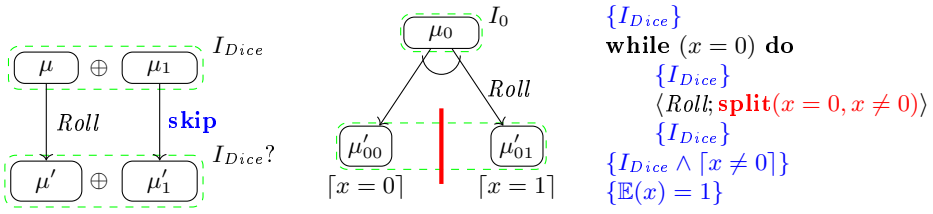
Example: a shared three-sided dice. To see the problem with branch statements more concretely, we consider a simple program $\mathbb{C}_{Dice}$ of n threads, where the code of each thread is *Dice*:

$$\mathit{Dice} \stackrel{\text{def}}{=} \mathbf{while}\ (x = 0)\ \mathbf{do}\ \mathit{Roll}, \quad \text{where}\ \ \mathit{Roll} \stackrel{\text{def}}{=} \left(x \stackrel{\$}{:=} \left\{1 : \tfrac{1}{2} \mid 2x : \tfrac{1}{6} \mid \tfrac{x}{2} : \tfrac{1}{3}\right\}\right)$$

Here x is a shared variable initialized to 0. The loop body *Roll* is a random assignment, which is short for the atomic probabilistic choice $\langle\langle x := 1\rangle \oplus_{\frac{1}{2}} (\langle x := 2x\rangle \oplus_{\frac{1}{3}} \langle x := \frac{x}{2}\rangle)\rangle$. That is, the thread atomically rolls a 3-sided dice and updates x according to the outcome: it sets x to 1 with probability $\frac{1}{2}$, doubles x with probability $\frac{1}{6}$ and halves x with probability $\frac{1}{3}$.

We want to verify that $\mathbb{C}_{Dice}$ satisfies the postcondition $\mathbb{E}(x) = 1$. As we explained, to do thread-local reasoning, we first find out the invariant I_{Dice} to model the interference:

$$I_{Dice} \stackrel{\text{def}}{=} I_0 \oplus I_1, \quad \text{where}\ \ I_0 \stackrel{\text{def}}{=} \lceil x = 0 \rceil \ \ \text{and}\ \ I_1 \stackrel{\text{def}}{=} (\lceil x \neq 0 \rceil \wedge \mathbb{E}(x) = 1)$$



(a) from layer to layer (b) split (c) proof

Fig. 7: Executions of *Dice* and Its Proof with Split

It says, every whole state distribution μ (at every layer of an execution forest) is a mixture $\mu_0 \oplus \mu_1$ (formed by taking μ_0 with arbitrary probability and taking μ_1 with the remaining probability) in which μ_0 and μ_1 satisfy I_0 and I_1 respectively.

To check I_{Dice} is indeed an invariant, one may consider showing that I_{Dice} is preserved by *Roll*. However, even if I_{Dice} is preserved by *Roll* (which is indeed true), it is still unclear whether I_{Dice} is preserved layer by layer. Specifically, after executing *Roll*, we will reach a state distribution whose support contains both the states satisfying $x = 0$ and those satisfying $x \neq 0$. From the former, the thread will enter the next round of the loop; but from the latter, the thread will exit the loop and do the code after the loop (or **skip** if there is no subsequent code). Consequently, *Roll* may be executed "at the same time" with **skip**, as shown in Fig. 7a. What we need to prove is that I_{Dice} is preserved by a *mixture* of executing *Roll* and **skip** at the same layer.

However, it is difficult to design logic rules to compose the proofs of *Roll* and **skip** for their mixture, because *Roll* as the loop body is actually syntactically sequenced before **skip**, the code after the loop. We face a similar problem as the problem with the **if**-statement, as explained above.

3.5 Our Key Idea: Split

Instead of trying to reason about the mixture of the behaviors of different statements at the whole layer, we *split* the state distribution of the layer, and reason about the different statements separately. In detail, we introduce an auxiliary command $\mathbf{split}(b_1, \ldots, b_k)$. It divides the current state distribution μ into k disjoint parts $\mu_1, \ldots, \mu_k$, such that each smaller distribution μ_i satisfies $\lceil b_i \rceil$ and μ is their mixture $\mu_1 \oplus \ldots \oplus \mu_k$. In our abstract operational semantics the thread *non-deterministically* picks a μ_k and continues its execution. One can instrument the code being verified with proper **split** commands so that each μ_k corresponds to a different branch of a branch statement. Note that the **split** commands only affects the abstract semantics. In the concrete semnatics, **split** has no effect and can be viewed as a no-op.

With split, the invariant I no longer needs to specify the whole layer μ, but instead it specifies only the smaller distributions μ_k generated by split. This I must be preserved by the execution at every μ_k. For instance, if we instrument

split$(b, \neg b)$ before **if** (b) **then** C_1 **else** C_2, then it suffices to prove that I is preserved by the executions of C_1 and C_2 at distributions satisfying $\lceil b \rceil$ and $\lceil \neg b \rceil$ respectively.

Split is physical and irreversible. We do not provide any command to mix back the smaller distributions that result from split. Instead of directly verifying $\vdash_{\text{A}} \{P\}\mathbb{C}\{Q\}$, where $\mathbb{C}$ contains no **split** commands and thus Q holds at the whole leaf layer, we verify $\vdash_{\text{A}} \{P\}\mathbb{C}'\{Q\}$ for $\mathbb{C}'$ that results from instrumenting $\mathbb{C}$ with auxiliary **split** commands. Therefore Q needs to hold at every smaller distribution at the leaf layer. That said, we do provide the following logic rule to convert $\vdash_{\text{A}} \{P\}\mathbb{C}'\{Q\}$ back to $\vdash_{\text{A}} \{P\}\mathbb{C}\{Q\}$:

$$\frac{\vdash_{\text{A}} \{P\}\mathbb{C}'\{Q\} \quad \textbf{closed}(Q)}{\vdash_{\text{A}} \{P\}\textbf{RemoveSplit}(\mathbb{C}')\{Q\}} \ (\textsc{removesplit})$$

Here **RemoveSplit**$(\mathbb{C}')$ removes all the **split** commands from $\mathbb{C}'$, and **closed**(Q) (defined at the end of Sec. 3.2) allows us to re-establish Q at the mixture of smaller distributions that all satisfy Q. The subscript "A" in the judgement indicates that the reasoning is based on the *abstract* semantics.

Proof for the shared three-sided dice. To verify *Dice*, we split the state distributions so that the states at which the thread enters the next round of the loop and those at which the thread exits the loop are always separate. As such, the invariant I_{Dice} is revised to be a *disjunction*:

$$I_{Dice} \stackrel{\text{def}}{=} I_0 \vee I_1, \quad \text{where } I_0 \stackrel{\text{def}}{=} \lceil x = 0 \rceil \text{ and } I_1 \stackrel{\text{def}}{=} (\lceil x \neq 0 \rceil \wedge \mathbb{E}(x) = 1)$$

In contrast to the earlier $I_0 \oplus I_1$ which holds at a mixture, this new I_{Dice} holds at a state distribution μ satisfying *either* I_0 *or* I_1. If μ satisfies I_0, the thread enters the next round of the loop; otherwise it exits the loop.

We instrument the loop body with the **split** command, as shown in red color in Fig. 7c. This **split** command ensures that the new I_{Dice} is indeed an invariant. As the blue assertions indicate, if I_{Dice} holds before the loop body, which means either I_0 or I_1 holds, then I_{Dice} still holds after atomically executing *Roll* and **split**. In particular, as shown in Fig. 7b, if I_0 holds before the loop body, executing *Roll* gives us a state distribution satisfying $\lceil x = 0 \rceil \oplus \lceil x = 1 \rceil$, and then executing **split**$(x = 0, x \neq 0)$ (see the red vertical bar) results in two separate state distributions μ'_{00} satisfying $\lceil x = 0 \rceil$ and μ'_{01} satisfying $\lceil x = 1 \rceil$. Both μ'_{00} and μ'_{01} satisfy I_{Dice}. The full proof is given in Appendix G.1.

Logic rules for split and branch statements. Below we introduce our logic rules for **split**, **if**-statements and **while**-loops to show how the split mechanism works.

$$\frac{G \vdash_{\text{SQ}} \{I \wedge P\}C\{(I \wedge Q \wedge \lceil b_1 \rceil) \oplus \cdots \oplus (I \wedge Q \wedge \lceil b_k \rceil)\} \quad \cdots}{R, G, I \vdash \{P\}\langle C\rangle\ \textbf{split}(b_1, \ldots, b_k)\{(Q \wedge \lceil b_1 \rceil) \vee \ldots \vee (Q \wedge \lceil b_k \rceil)\}} \ (\textsc{atom-split})$$

As in the *Dice* example, **split** is usually inserted after and executed atomically with some code $\langle C\rangle$. As such, we provide the command $\langle C\rangle$ **split**$(b_1, \ldots, b_k)$,

which has the same meaning as $\langle C; \mathbf{split}(b_1, \ldots, b_k)\rangle$. The (ATOM-SPLIT) rule requires us to prove the $\vdash_{\text{SQ}}$ judgment, which reasons about C as sequential code, and ensures that the state distribution at the end is a mixture of smaller distributions satisfying $\lceil b_1 \rceil, \ldots, \lceil b_n \rceil$ respectively. Since **split** turns the big distribution into these smaller ones as separate parts, the postcondition of the conclusion is a disjunctive assertion. We can see that split essentially turns $\oplus$ into $\vee$. The disjunction can be the precondition of the subsequent **if** and **while** statements as required by the (COND) and (WHILE) rules below. Here we omit the side conditions which says that the pre/post-conditions are stable with respect to R and I. The definition of rely/guarantee conditions and stability will be explained in Sec. 5.1 and the complete rule will be presented in Sec. 5.2.

$$\frac{P_1 \Rightarrow \lceil b \rceil \quad P_2 \Rightarrow \lceil \neg b \rceil \quad R, G, I \vdash \{P_1\} C_1 \{Q\} \quad R, G, I \vdash \{P_2\} C_2 \{Q\} \quad \cdots}{R, G, I \vdash \{P_1 \vee P_2\} \mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2 \{Q\}}\ (\text{COND})$$

$$\frac{P_1 \Rightarrow \lceil b \rceil \qquad P_2 \Rightarrow \lceil \neg b \rceil \wedge Q \qquad R, G, I \vdash \{P_1\} C \{P_1 \vee P_2\} \qquad \cdots}{R, G, I \vdash \{P_1 \vee P_2\} \mathbf{while}\ (b)\ \mathbf{do}\ C \{Q\}}\ (\text{WHILE})$$

Our (COND) rule assumes that, before the **if**-statement, the state distributions have already been split into smaller distributions for executing the **then**- and **else**-branches separately. Therefore, the precondition is supposed to be the disjunction $P_1 \vee P_2$, where $P_1 \Rightarrow \lceil b \rceil$ and $P_2 \Rightarrow \lceil \neg b \rceil$. Recall that $\lceil b \rceil$ says b holds with probability 1, i.e. all the states in the support of the distribution satisfy b. So, $\lceil b \rceil \vee \lceil \neg b \rceil$ is not implied by $\lceil b \vee \neg b \rceil$. The latter holds always, but for the former to hold, we must do split first. Then the branches can be verified independently, as we do in classical Hoare logic.

Similarly, in the (WHILE) rule, the loop invariant is the disjunction $P_1 \vee P_2$. Resulting from a split, the part satisfying P_1 ensures that the loop always continues with its next round since $P_1 \Rightarrow \lceil b \rceil$, while the part satisfying P_2 terminates the loop as $P_2 \Rightarrow \lceil \neg b \rceil$. If the value of b is probabilistic and can be modified by the code before the loop and by the loop body C, one need to insert **split** before the loop *and* inside the loop body C, so that $P_1 \vee P_2$ holds before every round of the loop.

4 The Programming Language

The syntax of the language is defined in Fig. 8. The whole program $\mathbb{C}$ consists of n sequential threads. The statements C of each thread are mostly standard. The *atomic statements* $\langle C \rangle$ and *the probabilistic choices* $\langle C_1 \rangle \oplus_p \langle C_2 \rangle$ are explained in Sec. 3. For verification purposes, we also append the atomic statements with split statements to get $(\langle C \rangle\ sp)$ where sp is in the form of $\mathbf{split}(b_1, \ldots, b_k)$.

Below we give two operational semantics to the language. The concrete one follows the standard interleaving semantics and models program steps as *probabilistic* transitions over program states. *The split statements are ignored in this semantics.* That is, they are viewed as annotations for verification only and have no operational effects.

$$
\begin{array}{rrl}
(\mathit{Nat})\ n, k & \in & \mathbb{N} \qquad (\mathit{Real})\ p, r \in \mathbb{R} \qquad (\mathit{PVar})\ x \in \mathit{String} \\
(\mathit{Expr})\ e & ::= & n \mid x \mid e_1 + e_2 \mid e_1 - e_2 \mid e_1 * e_2 \mid \dots \\
(\mathit{Bexp})\ b & ::= & \mathrm{true} \mid \mathrm{false} \mid e_1 < e_2 \mid e_1 = e_2 \mid e_1 \leq e_2 \mid \neg b \mid b_1 \wedge b_2 \mid b_1 \vee b_2 \mid \dots \\
(\mathit{SplitInstr})\ sp & ::= & \mathbf{split}(b_1, \dots, b_k) \\
(\mathit{Stmt})\ C & ::= & \mathbf{skip} \mid x := e \mid C_1; C_2 \mid \mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2 \mid \mathbf{while}\ (b)\ \mathbf{do}\ C \\
 & & \mid \langle C \rangle \mid \langle C \rangle\ sp \mid \langle C_1 \rangle \oplus_p \langle C_2 \rangle \\
(\mathit{Prog})\ \mathbb{C} & ::= & C_1 \parallel \dots \parallel C_n
\end{array}
$$

Fig. 8: The Programming Language

Thread IDs, schedules, states and states distributions:

$$
\begin{array}{ll}
(\mathit{ThreadId})\ t \in \mathbb{N}_+ & (\mathit{Schedule})\ \varphi ::= t :: \varphi \quad \text{(coinductive)} \\
(\mathit{State})\ \sigma \in \mathit{PVar} \to \mathbb{R} & (\mathit{DState})\ \mu \in \mathbb{D}_{\mathit{State}}
\end{array}
$$

Global transitions: $(\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')$

$$
\frac{(C_t, \sigma) \xrightarrow{p} (C_t', \sigma')}{(C_1 \parallel \dots \parallel C_t \parallel \dots \parallel C_n, \sigma) \xrightarrow[t]{p} (C_1 \parallel \dots \parallel C_t' \parallel \dots \parallel C_n, \sigma')}
$$

Thread-local transitions: $(C, \sigma) \xrightarrow{p} (C', \sigma')$

$$
\frac{[\![e]\!]_\sigma = n}{(x := e, \sigma) \xrightarrow{1} (\mathbf{skip}, \sigma\{x \rightsquigarrow n\})} \qquad \frac{}{(\mathbf{skip}, \sigma) \xrightarrow{1} (\mathbf{skip}, \sigma)}
$$

$$
\frac{C_1 \neq \mathbf{skip} \quad (C_1, \sigma) \xrightarrow{p} (C_1', \sigma')}{(C_1; C_2, \sigma) \xrightarrow{p} (C_1'; C_2, \sigma')} \qquad \frac{}{(\mathbf{skip}; C_2, \sigma) \xrightarrow{1} (C_2, \sigma)}
$$

$$
\frac{}{(\langle C_1 \rangle \oplus_p \langle C_2 \rangle, \sigma) \xrightarrow{p} (\langle C_1 \rangle, \sigma)} \qquad \frac{}{(\langle C_1 \rangle \oplus_p \langle C_2 \rangle, \sigma) \xrightarrow{1-p} (\langle C_2 \rangle, \sigma)}
$$

$$
\frac{\exists k. \forall n \geq k. (C, \sigma) \xrightarrow{p}{}^{n} (\mathbf{skip}, \sigma')}{(\langle C \rangle, \sigma) \xrightarrow{p} (\mathbf{skip}, \sigma')} \qquad \frac{(\langle C \rangle, \sigma) \xrightarrow{p} (\mathbf{skip}, \sigma')}{(\langle C \rangle\ \mathbf{split}(b_1, \dots, b_k), \sigma) \xrightarrow{p} (\mathbf{skip}, \sigma')}
$$

Fig. 9: Concrete Operational Semantics

The abstract semantics models program steps as transitions over distributions of program configurations. We also assign operational semantics to **split** statements. We prove that Hoare-triples valid in the abstract semantics are also valid in the concrete semantics (Thm 1 below).

4.1 Concrete Operational Semantics

We show selected semantics rules in Fig. 9 and give the full set of rules in Appendix B. The single-step transition of the whole program is defined through the thread-local transitions. Each step is decorated with a p, the probability that the step may occur. For most thread-local transitions except the probabilistic choices and atomic statements, p is simply 1. Note that we allow the **skip** command at the end of execution to stutter with probability 1, but it cannot stutter if it is

sequenced before some C. That is, "**skip**; C" can only step to C. $\langle C_1\rangle \oplus_p \langle C_2\rangle$ chooses to execute the left or right branches, with probability p and $1-p$, respectively. The atomic statement $\langle C\rangle$ is always done in one step, no matter how complicated C is. We assume C in the atomic statement never contains **while**-loops, so it always terminates in a bounded number of steps. Note that the need of atomicity of the branches in $\langle C_1\rangle \oplus_p \langle C_2\rangle$ is not overly idealistic, because we mainly use $\langle C_1\rangle \oplus_p \langle C_2\rangle$ to encode a random assignment, thus C_1 and C_2 themselves may correspond to single instructions at the machine level anyway (in this case, the atomic wrappers $\langle\cdot\rangle$ are unnecessary). In the proofs of algorithms, we may insert auxiliary statements (a.k.a. ghost code) to be executed with the probabilistic choice together in one step. This is actually the only case when C_1 or C_2 is non-atomic and needs to be wrapped by $\langle\cdot\rangle$. The more general form of $C_1 \oplus_p C_2$ can be encoded as $\langle x := \text{true}\rangle \oplus_p \langle x := \text{false}\rangle$; **if** (x) **then** C_1 **else** C_2.

Before giving semantics to $\langle C\rangle$, we first introduce the n-step thread-local transition, represented as $(C,\sigma) \xrightarrow{p}{}^{n} (C',\sigma')$. Informally, if there is only one n-step execution path from (C,σ) to (C',σ'), the probability p in $(C,\sigma) \xrightarrow{p}{}^{n} (C',\sigma')$ is the product of the probability of every step on the path. If there are more than one execution paths, we need to sum up the probabilities of all the paths. We give an example to illustrate the n-step thread-local transition in Appendix. B

Then the operational semantics rule for $\langle C\rangle$ says it finishes the execution of C in one step (that is, the execution of C cannot be interrupted by other threads). Note that $\langle C\rangle$ may lead to different states with different probabilities, since C may contain probabilistic choices.

The multi-step transition $((\mathbb{C},\sigma) \xrightarrow[\varphi]{p}{}^{n} (\mathbb{C}'',\sigma''))$ of the whole program $\mathbb{C}$ under the schedule φ is similar to the multi-step thread-local transitions. The schedule φ is an infinite sequence of thread IDs. It decides which thread t is to be executed next. The accumulated probability of an n-step transition is the *sum* of the probability of every possible execution path.

Below we define $[\![\mathbb{C}]\!]_\varphi$ as a function that maps an *initial state* σ to a sub-distribution of *final states*. We also lift the function to the distribution μ of the initial states.

$$[\![\mathbb{C}]\!]_\varphi(\sigma) \stackrel{\text{def}}{=} \lambda\sigma'.\ \lim \vec{p}_{\sigma'},\quad \text{where } \forall n.\ (\mathbb{C},\sigma) \xrightarrow[\varphi]{\vec{p}_{\sigma'}[n]}{}^{n} (\textbf{skip} \parallel \cdots \parallel \textbf{skip}, \sigma')$$

$$[\![\mathbb{C}]\!]_\varphi(\mu) \stackrel{\text{def}}{=} \mathbb{E}_{\sigma\sim\mu}\{[\![\mathbb{C}]\!]_\varphi(\sigma)\} \qquad \text{(see Eqn. 7 for the expected sub-distribution)}$$

Here $\vec{p}_{\sigma'}$ is an infinite sequence of probabilities and $\vec{p}_{\sigma'}[n]$ is the n-th element of the sequence. Note $\lim \vec{p}_{\sigma'}$ always exists as we can prove $\vec{p}_{\sigma'}$ always converges.

Then we can give a simple definition of the partial correctness of $\mathbb{C}$ with respect to the precondition P and the postcondition Q, which are assertions over state distributions and are defined in Sec. 5.1.

Definition 9. $\models \{P\}\mathbb{C}\{Q\}$ iff, for all μ and φ, if $\mu \models P$, and $|[\![\mathbb{C}]\!]_\varphi(\mu)| = 1$, then $[\![\mathbb{C}]\!]_\varphi(\mu) \models Q$.

The premise $|[\![\mathbb{C}]\!]_\varphi(\mu)| = 1$ requires the execution of $\mathbb{C}$ (with the schedule φ and the initial state distribution μ) *terminates with probability* 1.

$$
\begin{array}{rll}
W & \in \mathbb{D}_{Prog\times State} & W|_b \stackrel{\text{def}}{=} W|_{\lambda(\mathbb{C},\sigma).\sigma\models b} \\
\delta(\mathbb{C}) & \stackrel{\text{def}}{=} \lambda\mathbb{C}_1. \begin{cases} 1, & \text{if } \mathbb{C}_1 = \mathbb{C} \\ 0, & \text{otherwise} \end{cases} & \\
\mathit{init}(\mathbb{C},\mu) & \stackrel{\text{def}}{=} \delta(\mathbb{C}) \otimes \mu \quad \text{(see Eqn. 4 for the definition of } \otimes) & \\
\mathit{nextsplit}(C) & \stackrel{\text{def}}{=} \begin{cases} \mathbf{split}(b_1,\ldots,b_k), & \text{if } C = \langle C_1\rangle\ \mathbf{split}(b_1,\ldots,b_k) \\ \mathit{nextsplit}(C_1), & \text{if } C = C_1; C_2 \\ \mathbf{split}(\text{true}), & \text{otherwise} \end{cases} & \\
\mathit{nextsplit}(W,t) & \stackrel{\text{def}}{=} \{\mathit{nextsplit}(C_t) \mid (C_1 \parallel \cdots \parallel C_n, \sigma) \in \mathit{supp}(W)\} &
\end{array}
$$

$$
W \overset{t}{\rightsquigarrow} W' \quad \text{iff } W' = \lambda(\mathbb{C}',\sigma').\textstyle\sum_{\mathbb{C},\sigma}\{p \cdot W(\mathbb{C},\sigma) \mid (\mathbb{C},\sigma) \xrightarrow[t]{p} (\mathbb{C}',\sigma')\}
$$

$$
\frac{W \overset{t}{\rightsquigarrow} W' \quad \mathit{nextsplit}(W,t) = \{\mathbf{split}(b_1,\ldots,b_k)\} \quad W'|_{b_i} = W''}{W \overset{t}{\hookrightarrow} W''}
$$

$$
\frac{W \overset{t}{\rightsquigarrow} W' \quad \#\mathit{nextsplit}(W,t) > 1}{W \overset{t}{\hookrightarrow} W'}
$$

Fig. 10: Abstract Operational Semantics

4.2 Abstract Operational Semantics

The abstract semantics, shown in Fig. 10, models each step as a transition between distributions W of the whole program configurations $(\mathbb{C},\sigma)$. Also we give semantics to **split** statements.

Below we use $\mathit{nextsplit}(W,t)$ to represent the set consisting of the next **split** statements to be executed in the thread t of the program configurations in $\mathit{supp}(W)$. The next **split** statement of the thread t is sp if the next statement to be executed is in the form of $\langle C\rangle\ sp$, otherwise the next split is defined as $\mathbf{split}(\text{true})$. Throughout this paper, we assume all the splits $\mathbf{split}(b_1,\ldots,b_k)$ satisfy the following validity check, which says for any state there is always one and only one b_i that holds.

Definition 10. A split statement is valid, i.e., $\mathbf{validsplit}(\mathbf{split}(b_1,\ldots,b_k))$ holds, if and only if for any state σ, $\forall i,j.\ i \neq j \implies \sigma \models \neg(b_i \wedge b_j)$ and $\sigma \models b_1 \vee \ldots \vee b_k$.

The transition $W \overset{t}{\hookrightarrow} W''$ is done in two steps. First we make the transition $W \overset{t}{\rightsquigarrow} W'$ based on the concrete semantics, without considering splits. Then the splits in $\mathit{nextsplit}(W,t)$ are executed. We expect $\mathit{nextsplit}(W,t)$ to be a singleton set, i.e. threads t in different program configurations in $\mathit{supp}(W)$ all have the same subsequent **split** statement. We non-deterministically pick b_i from $b_1\ \ldots b_k$, and let W'' be the filtered distribution $W'|_{b_i}$ (see Fig. 10 and Eqn. 2 for the definition of $W|_b$). If the **split** statement is $\mathbf{split}(\text{true})$, we know W'' is the same as W'. If $\mathit{nextsplit}(W,t)$ contains more than one **split** statements, then we view the program as inappropriately instrumented. In this case we ignore all the split statements in $\mathit{nextsplit}(W,t)$ and let W'' be W'.

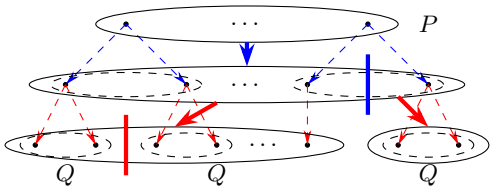


Fig. 11: Illustration of $\models_{\text{A}} \{P\}\mathbb{C}\{Q\}$

Figure 11 illustrates the execution. The dashed arrows represent state transitions in the concrete semantics, while the solid arrows represent the transitions $W \overset{t}{\rightsquigarrow} W'$ in the abstract semantics. Like before, we use different colors to represent actions of different threads. The vertical bars represent splits. The solid arrow and the split together correspond to the transition $W \overset{t}{\hookrightarrow} W''$. The branching shown by the two solid red arrows reflects the *non-deterministic* choice of the cases of the split.

Before giving the partial correctness under the abstract semantics, we first define the termination of W_0 in Def. 11: if the execution sequence of W_0 under the abstract semantics converges with the limit W, we say W_0 terminates at W. Here $\mathbf{History}(W_0, \varphi, \vec{W})$ says that $\vec{W}$ is a possibly infinite sequence $W_0, W_1, \ldots$ where $W_i \overset{\varphi[i]}{\hookrightarrow} W_{i+1}$ for every i. The formal definition of **History** can be found in Appendix B. The limit $(\lim \vec{W})$ is defined by Def. 6. The projection of W over code ($W^{(Prog)}$) and state ($W^{(State)}$) are defined by Eqn. 5.

Definition 11 (Termination of W). Given W_0 and a schedule φ. We say W_0 *terminates at* W *under the schedule* φ, represented as $W_0 \Downarrow_\varphi W$, if and only if there is an infinite sequence $\vec{W}$ such that $\mathbf{History}(W_0, \varphi, \vec{W})$, $\lim \vec{W} = W$ and $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$.

Next we define the partial correctness under the abstract semantics, $\models_{\text{A}} \{P\}\mathbb{C}\{Q\}$. The initial distribution of program configurations is $\mathit{init}(\mathbb{C}, \mu)$. As defined in Fig. 10, $\mathit{init}(\mathbb{C}, \mu)$ says the initial program is always $\mathbb{C}$ and the state distribution is μ. Figure 11 illustrates the meaning of $\models_{\text{A}} \{P\}\mathbb{C}\{Q\}$: if P holds over the initial distribution, Q must hold over *every* final distribution. Theorem 1 shows that the partial correctness in the abstract semantics implies the partial correctness in the concrete semantics when the postcondition is closed. Below we develop our program logic based on this abstract semantics.

Definition 12. $\models_{\text{A}} \{P\}\mathbb{C}\{Q\}$ iff for all μ, if $\mu \models P$, then for all φ and W, if $\mathit{init}(\mathbb{C}, \mu) \Downarrow_\varphi W$, then $W^{(State)} \models Q$.

Theorem 1. For all $P, \mathbb{C}, Q$, if $\models_{\text{A}} \{P\}\mathbb{C}\{Q\}$ and $\mathbf{closed}(Q)$, then $\models \{P\}\mathbb{C}\{Q\}$.

5 The Program Logic

We present the assertion language and the logic rules in this section.

$$
\begin{array}{rrl}
(\mathit{Assertion}) & \mathbf{p},\mathbf{q} & ::= b \mid \neg\mathbf{q} \mid \mathbf{q}_1 \wedge \mathbf{q}_2 \mid \mathbf{q}_1 \vee \mathbf{q}_2 \mid \forall X.\mathbf{q} \mid \exists X.\mathbf{q} \mid \ldots \\
(\mathit{Pexp}) & \xi & ::= r \mid \mathbb{E}(e) \mid \mathbf{Pr}(\mathbf{q}) \mid \xi_1 + \xi_2 \mid \xi_1 - \xi_2 \mid \xi_1 * \xi_2 \mid \ldots \\
(\mathit{PAssertion}) & P,Q,M,I & ::= \lceil \mathbf{q} \rceil \mid \xi_1 < \xi_2 \mid \xi_1 = \xi_2 \mid \xi_1 \leq \xi_2 \mid \neg Q \mid Q_1 \wedge Q_2 \mid Q_1 \vee Q_2 \\
 & & \quad \mid \forall X.Q \mid \exists X.Q \mid Q_1 \oplus_p Q_2 \mid Q_1 \oplus Q_2 \mid \ldots \\
(\mathit{Action}) & R,G & ::= \mathbf{p} \ltimes \mathbf{q} \mid [\mathbf{q}] \mid \neg R \mid R_1 \wedge R_2 \mid R_1 \vee R_2 \mid \forall X.R \mid \exists X.R \mid R_1 \circ R_2 \mid \ldots
\end{array}
$$

Fig. 12: The Assertion Language

Evaluation of probabilistic expressions:

$$[\![\mathbb{E}(e)]\!]_\mu \stackrel{\text{def}}{=} \mathbb{E}_{\sigma\sim\mu}[[\![e]\!]_\sigma] \qquad [\![\mathbf{Pr}(\mathbf{q})]\!]_\mu \stackrel{\text{def}}{=} \mathbf{Pr}_{\sigma\sim\mu}[\sigma \models \mathbf{q}]$$

Semantics of probabilistic assertions:

$\mu \models \lceil \mathbf{q} \rceil$ iff for all $\sigma \in \mathit{supp}(\mu)$, $\sigma \models \mathbf{q}$

$\mu \models Q_1 \oplus_p Q_2$ iff $p = 1$ and $\mu \models Q_1$, or $p = 0$ and $\mu \models Q_2$, or $0 < p < 1$ and there exists μ_1 and μ_2 such that $\mu = \mu_1 \oplus_p \mu_2$, $\mu_1 \models Q_1$ and $\mu_2 \models Q_2$

$\mu \models Q_1 \oplus Q_2$ iff there exists p such that $\mu \models Q_1 \oplus_p Q_2$

Fig. 13: Semantics of Assertions

5.1 The Assertion Language

We show the syntax of assertions in Fig. 12 and their semantics in Fig. 13. We use $\mathbf{p}$ and $\mathbf{q}$ to represent classical assertions over states, and P, Q and I for *probabilistic assertions* over state distributions. We also use ξ to denote *probabilistic expressions* such as the expected value of an arithmetic expression or the probability of a classical assertion. The expression ξ evaluates to a real number under the state distribution μ, represented as $[\![\xi]\!]_\mu$. $\mathbb{E}(e)$ evaluates to the expected value of $[\![e]\!]_\sigma$ (where $\sigma \in \mathit{supp}(\mu)$). $\mathbf{Pr}(\mathbf{q})$ evaluates to the probability of $\sigma \models \mathbf{q}$ (where $\sigma \in \mathit{supp}(\mu)$). The key definitions of expected values and probability of assertions are shown in Eqn. (1).

The assertion $\lceil \mathbf{q} \rceil$ lifts the state assertion $\mathbf{q}$ to a probabilistic assertion. It says $\mathbf{q}$ holds on all states in the support of the state distribution. The assertion $P \oplus_p Q$ holds at μ, if μ is a *mixture* of two distributions μ_0 and μ_1, which are associated with probabilities p and 1–p, and satisfy P and Q respectively. $Q_1 \oplus Q_2$ says there exists p such that $Q_1 \oplus_p Q_2$ holds. The semantics of $\forall X.Q$ and $\exists X.Q$ are given in Appendix D. Throughout this paper, we use capital letters X to indicate that X is a logical variable and lowercase letters x to indicate that x is a program variable. We define **true** as a syntactic sugar of $\lceil \mathrm{true} \rceil$ which holds on all state distributions.

Actions R and G are assertions over state transitions. Their semantics, $(\sigma, \sigma') \models R$, is the same as that in classical (non-probabilistic) rely-guarantee logics. We use $[\![R]\!]$ to denote the set of state transitions that satisfy R.

Stability We define the stability of a probabilistic assertion Q with respect to the environment interference (specified by I and R) in Fig. 14. We first define

$$\mu \overset{R}{\mapsto} \mu' \quad \text{iff } \exists \theta \in \mathcal{P}(State \times State).\, \theta \subseteq [\![R]\!] \wedge supp(\mu) = dom(\theta) \wedge supp(\mu') = range(\theta)$$

$$\mu \underset{I}{\overset{R}{\rightarrowtail}} \mu'' \quad \text{iff } \mu \models I \wedge (\exists \mu'.\, \mu \overset{R}{\mapsto} \mu' \wedge supp(\mu'') \subseteq supp(\mu')) \wedge \mu'' \models I$$

$$\mathbf{Sta}(Q, R, I) \text{ iff } \forall \mu, \mu'.\, \mu \models Q \wedge \mu \underset{I}{\overset{R}{\rightarrowtail}} \mu' \implies \mu' \models Q$$

Fig. 14: Stability

$\mu \underset{I}{\overset{R}{\rightarrowtail}} \mu''$ to describe that the current state distribution is changed from μ to μ'' due to the environment interference. As we can see in the abstract operational semantics, every transition made by a thread is done in two steps. The first step is normal execution without splits and the second step is the execution of **split**. Similarly, we model the execution of the environment in two steps. The first step is $\mu \overset{R}{\mapsto} \mu'$. It requires us to find a set θ of state transitions allowed by R (i.e. $\theta \subseteq [\![R]\!]$), such that θ transforms the states of $supp(\mu)$ to those of $supp(\mu')$. The second step is the execution of **split** statements by the environment. The condition $supp(\mu'') \subseteq supp(\mu')$ abstracts the behaviors of **split**. In addition, the environment needs to preserve the invariant I, so $\mu \models I \wedge \mu'' \models I$. Then we can give a simple definition of $\mathbf{Sta}(Q, R, I)$ in Fig. 14.

In general, it is not easy to prove the stability of a probabilistic assertion with respect to classical rely conditions. But in practice, the thread-local pre/post-conditions and intermediate assertions P are usually "non-probabilistic", in the form of $\lceil b_1 \rceil \vee \ldots \vee \lceil b_n \rceil$. This is because the probabilistic information is often about the shared resource and has already been specified by the global invariant I. For such P, proving stability $\mathbf{Sta}(P, R, I)$ is not much harder than proving stability in the classical rely-guarantee reasoning. We give some rules to syntactically proving $\mathbf{Sta}(P, R, I)$ in Appendix D.

Closed Assertions As explained in Sec. 3.5, we need the postcondition of the whole program to be closed for applying split. $\mathbf{closed}(Q)$ means that the mixture of any (maybe countably infinite) number of state distributions satisfies Q if each of them satisfies Q.

Definition 13. An assertion Q is closed, i.e., $\mathbf{closed}(Q)$ holds, if and only if, for all $\nu \in \mathbb{D}_{\mathbb{D}_{State}}$, if $\mu \models Q$ holds for all $\mu \in supp(\nu)$, then $\overline{\nu} \models Q$ (see Eqn. (8) for the definition of $\overline{\nu}$).

Many assertions are closed, such as $\lceil x = 1 \rceil$, $\mathbf{Pr}(y > 2) = 0.5$, $\lceil x = 0 \rceil \oplus \lceil x = 1 \rceil$. We give syntactic rules in Appendix D to prove closedness of assertions. There do exist non-closed assertions, such as $\lceil x = 1 \rceil \vee \lceil x = 2 \rceil$ and $\mathbf{Pr}(x = 0) \neq 0.5$. In this work, we focus on the class of randomized algorithms whose correctness is about the bound of the probability of a random event or the expected value of a random variable. For this kind of algorithms, our syntactic rules for closedness are useful enough.

Limit-Closed Assertions To verify almost surely terminating programs, we require the invariant I and the postconditions of all threads are limit-closed assertions. Below we define limit-closed assertions (see Def. 6 for the definition of $\lim \vec{\mu}$).

Definition 14. An assertion Q is limit-closed, i.e., $\mathbf{lclosed}(Q)$ holds, if and only if, for all infinite sequences $\vec{\mu}$, if $\lim \vec{\mu} = \mu$, and $\vec{\mu}[n] \models Q$ holds for all n, then $\mu \models Q$.

We also give syntactic rules in Appendix D to prove that an assertion is limit-closed. They are similar to those for closedness and thus are also useful in verifying algorithms whose correctness is about the bound of the probability of a random event or the expected value of a random variable.

5.2 Inference Rules

Our inference rules are organized into three layers for the whole program, the thread local reasoning, and sequential reasoning, as shown in Fig. 15. The top-level judgement for the whole program is in the form of $\vdash_{\text{A}} \{P\}\mathbb{C}\{Q\}$ where "A" means abstract. One can use the parallel composition rule (PAR) to decompose the verification of concurrent programs into the verification of each thread. The judgement for thread-local reasoning is in the form of $R, G, I \vdash \{P\}C\{Q\}$ where R and G are rely/guarantee conditions and I is the layer invariant. To verify atomic blocks, one can use the (ATOM) and (ATOM-SPLIT) rules to apply sequential reasoning to the code in the atomic blocks. The judgement for sequential reasoning is in the form of $\vdash_{\text{SQ}} \{P\}C\{Q\}$ where "SQ" means sequential.

Whole-Program Rules The top-level rules are used to verify whole programs. The judgement is in the form of $\vdash_{\text{A}} \{P\}\mathbb{C}\{Q\}$. Here P and Q are probabilistic assertions, which specify the initial state distributions and the terminating state distributions respectively.

The parallel composition rule (PAR) is (mostly) standard. The invariant I and the postcondition of each thread $Q_1, \ldots, Q_n$ are required to be limit-closed assertions, which ensures that the limit state distribution of the infinite sequence produced by $\mathbb{C}$ under the abstract operational semantics satisfies I and $Q_1, \ldots, Q_n$.

The (LAZYCOIN) rule is used to verify probabilistic choices. Note that the execution of $\langle C_1 \rangle \oplus_p \langle C_2 \rangle$ is *not* atomic, and its two steps (i.e. the coin flip and the execution of $\langle C_1 \rangle$ or $\langle C_2 \rangle$) can interleave with the environment steps. The (LAZYCOIN) rule allows us to verify $\mathbf{lazycoin}(\mathbb{C})$ instead of $\mathbb{C}$, where $\mathbf{lazycoin}(\mathbb{C})$ replaces every $\langle C_1 \rangle \oplus_p \langle C_2 \rangle$ in $\mathbb{C}$ with $\mathbf{skip}; \langle \langle C_1 \rangle \oplus_p \langle C_2 \rangle \rangle$. We can view **lazycoin** as a transformation that defers the coin flip step to be executed with $\langle C_1 \rangle$ or $\langle C_2 \rangle$ together. This transformation is sound because, in the OA model, the scheduler and the environment threads should not be aware of the outcome of the coin flip, so we can soundly swap the coin-flip step and the environment steps, and reason about the atomic probabilistic choice $\langle \langle C_1 \rangle \oplus_p \langle C_2 \rangle \rangle$ instead. The extra **skip** is to ensure that the new code has the same number of steps as the non-atomic

Whole program rules: $\vdash_{\text{A}} \{P\}\mathbb{C}\{Q\}$

$$\frac{\forall i.\ R_i, G_i, I \vdash \{P_i\}C_i\{Q_i\} \qquad \forall i,j.\ i \neq j \implies (G_i \Rightarrow R_j) \quad P \Rightarrow I \wedge P_1 \wedge \cdots \wedge P_n \quad I \wedge Q_1 \wedge \cdots \wedge Q_n \Rightarrow Q \quad \mathbf{lclosed}(\{I, Q_1, \ldots, Q_n\})}{\vdash_{\text{A}} \{P\}C_1 \parallel \cdots \parallel C_n\{Q\}}\ (\textsc{par})$$

$$\frac{\vdash_{\text{A}} \{P\}\mathbb{C}\{Q\} \quad \mathbf{closed}(Q)}{\vdash_{\text{A}} \{P\}\mathbf{RemoveSplit}(\mathbb{C})\{Q\}}\ (\textsc{removesplit}) \qquad \frac{\vdash_{\text{A}} \{P\}\mathbf{lazycoin}(\mathbb{C})\{Q\}}{\vdash_{\text{A}} \{P\}\mathbb{C}\{Q\}}\ (\textsc{lazycoin})$$

Thread-local rules: $R, G, I \vdash \{P\}C\{Q\}$

$$\frac{G \vdash_{\text{SQ}} \{I \wedge P\}C\{I \wedge Q\} \quad \mathbf{Sta}(\{P, Q\}, R, I)}{R, G, I \vdash \{P\}\langle C\rangle\{Q\}}\ (\textsc{atom}) \qquad \frac{R, G, I \vdash \{P\}C_1\{M\} \quad R, G, I \vdash \{M\}C_2\{Q\}}{R, G, I \vdash \{P\}C_1; C_2\{Q\}}\ (\textsc{seq})$$

$$\frac{G \vdash_{\text{SQ}} \{I \wedge P\}C\{(I \wedge Q \wedge \lceil b_1\rceil) \oplus \cdots \oplus (I \wedge Q \wedge \lceil b_k\rceil)\} \quad \mathbf{Sta}(\{P, Q \wedge (\lceil b_1\rceil \vee \cdots \vee \lceil b_k\rceil)\}, R, I)}{R, G, I \vdash \{P\}\langle C\rangle\ \mathbf{split}(b_1, \ldots, b_k)\{(Q \wedge \lceil b_1\rceil) \vee \ldots \vee (Q \wedge \lceil b_k\rceil)\}}\ (\textsc{atom-split})$$

$$\frac{P_1 \Rightarrow \lceil b\rceil \quad P_2 \Rightarrow \lceil \neg b\rceil \quad \mathbf{Sta}(P_1 \vee P_2, R, I) \quad R, G, I \vdash \{P_1\}C_1\{Q\} \quad R, G, I \vdash \{P_2\}C_2\{Q\}}{R, G, I \vdash \{P_1 \vee P_2\}\mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2\{Q\}}\ (\textsc{cond})$$

$$\frac{P_1 \Rightarrow \lceil b\rceil \quad P_2 \Rightarrow \lceil \neg b\rceil \wedge Q \quad R, G, I \vdash \{P_1\}C\{P_1 \vee P_2\} \quad \mathbf{Sta}(\{P_1 \vee P_2, Q\}, R, I)}{R, G, I \vdash \{P_1 \vee P_2\}\mathbf{while}\ (b)\ \mathbf{do}\ C\{Q\}}\ (\textsc{while})$$

Fig. 15: Selected Logic Rules

$\langle C_1\rangle \oplus_p \langle C_2\rangle$, and thus to ensure that $\mathbf{lazycoin}(\mathbb{C})$ and $\mathbb{C}$ generate the same behaviors in the OA model. Note that (LAZYCOIN) is *unsound* in the SA model.

The (REMOVESPLIT) rule has been explained in Sec. 3. We also support the standard consequence rule, conjunction rule and disjunction rule for whole programs, which are shown in Appendix. E.

Thread-Local Rules The thread-local judgement is in the form of $R, G, I \vdash \{P\}C\{Q\}$. The rely/guarantee conditions R and G are non-probabilistic and their meaning are the same as in the traditional rely-guarantee reasoning. The invariant I specifies the probabilistic property that is preserved by both the thread and its environment at every layer. The rely/guarantee conditions need to be reflexive in well-formed thread-local judgements.

To verify $\langle C\rangle$, the (ATOM) rule verifies C as sequential code, and requires I is preserved at the end if it holds at the beginning, and the whole state transitions resulting from the sequential execution C satisfy the guarantee G. The pre/post-conditions need to be stable with respect to R and I. We use $\mathbf{Sta}(\{P, Q\}, R, I)$ as a shorthand for $\mathbf{Sta}(P, R, I) \wedge \mathbf{Sta}(Q, R, I)$. Similar representations are used in the remaining part of the paper.

Our (SEQ) rule for sequential composition is standard. The (ATOM-SPLIT), (COND) and (WHILE) rules have been explained in Sec. 3.5. Note that (ATOM-

SPLIT) cannot be replaced by (ATOM), since only **split** can turn $\oplus$ into $\vee$ (see the first premise and conclusion's postconditions in (ATOM-SPLIT)).

Sequential Rules The judgement for sequential rules is in the form of $G \vdash_{\text{SQ}} \{P\}C\{Q\}$. Note that the guarantee G does not specify the state transition of every single step of C. Instead it specifies the state transitions from initial states to the corresponding final states at the end of C. The rules for sequential reasoning are simple extensions of those in [6] and are presented in Appendix E.

Soundness The following theorem shows that our logic is sound with respect to the abstract operational semantics, where $\models_{\text{A}} \{P\}\mathbb{C}\{Q\}$ is given in Def. 12.

Theorem 2. *For all* $P, \mathbb{C}, Q$, *if* $\vdash_{\text{A}} \{P\}\mathbb{C}\{Q\}$, *then* $\models_{\text{A}} \{P\}\mathbb{C}\{Q\}$.

6 Case Study: Conciliator

As introduced in Sec. 1, Chor et al. [12] give a probabilistic-write based conciliator for *probabilistic agreement* between n threads, each thread i executing C_i below, where s is a shared variable and y_i is the local variable for thread i that records its return value.

$$C_i \stackrel{\text{def}}{=} (\textbf{while } (s = 0) \textbf{ do } \langle s := i\rangle \oplus_p \langle \textbf{skip}\rangle)\,; y_i := s$$

We want to prove $\{\lceil s = 0\rceil\}C_1 \parallel \cdots \parallel C_n\{\mathbf{Pr}(y_1 = \cdots = y_n) \geq (1-p)^{n-1}\}$. Intuitively the postcondition holds because, when there is exactly one thread i which succeeds in writing to s, all threads will return i. This ideal case happens with probability no less than $(1-p)^{n-1}$ in OA, because (i) for the program to terminate, at least one thread has updated s, and (ii) after the first update to s, each of the other $n-1$ threads has at most one chance to update s, and such an update happens with probability no more than $1-p$. Note that this algorithm does *not* work in SA, where different threads can be scheduled for different outcomes of coin flips, making the aforementioned ideal case happens with probability less than $(1-p)^{n-1}$.

To formulate the intuition, we introduce a shared auxiliary variable c that counts how many threads have updated s and insert the auxiliary code $c := c+1$ which is executed atomically with $s := i$. We also introduce flag variables d_i to formalize the "at most one chance" update to s. When d_i is set, it means thread i can no longer update s. We insert the auxiliary code $\mathit{SetFlag}_i$ to set d_i at the proper time. At the whole-program level, we apply (LAZYCOIN) and (REMOVESPLIT) to wrap the probabilistic choice in an atomic block, and to instrument $\textbf{split}(s = 0, s \neq 0)$ at the end of the loop body such that the resulting smaller distributions either enter or exit the loop, respectively. Using the (PAR) rule, our goal becomes to thread-locally verify the code below.

$$(\textbf{while } (s = 0) \textbf{ do } (\textbf{skip}; \langle \mathit{PWrite}_i\rangle\ \textbf{split}(s = 0, s \neq 0)\ ))\,; \langle\ \mathit{SetFlag}_i\ ; y_i := s\rangle,$$

where $\mathit{PWrite}_i \stackrel{\text{def}}{=} \langle s := i;\ c := c+1; \mathit{SetFlag}_i\rangle \oplus_p \langle\ \mathit{SetFlag}_i\ \rangle$

and $\mathit{SetFlag}_i \stackrel{\text{def}}{=} \textbf{if } (s \neq 0) \textbf{ then } d_i := 1 \textbf{ else skip}$

We define the invariant I below, which says that either $s = 0$ (and thus $c = 0$ and each thread has chance to update s), or $s \neq 0$ (and thus $c > 0$) and the probability of $c = 1$ has a lower bound.

$$I \stackrel{\text{def}}{=} I_0 \vee I_1, \text{ where } I_0 \stackrel{\text{def}}{=} \lceil s = 0 \wedge c = 0 \wedge \forall i.\, d_i = 0 \rceil,\ I_1 \stackrel{\text{def}}{=} \lceil s \neq 0 \wedge c > 0 \rceil \wedge \mathsf{PBound},$$
$$\text{and } \mathsf{PBound} \stackrel{\text{def}}{=} \exists K \leq n. \lceil \textstyle\sum_{i=1}^{n} d_i = K \rceil \wedge \mathbf{Pr}(c = 1) \geq (1-p)^{K-1}$$

We give the detailed proofs in Appendix G.2. The logic presented in the paper requires us to split in each round of the while-loop. This technique is sufficient to prove conciliator and *Dice*. However, for more advanced examples such as group election and multiplayer level-up game (in the TR), their loops require split in the first few rounds only. Thus, we extend the logic with a new while rule for while loops and a new sequential composition rule for sequential statements. With the two new rules, we can prove the two advanced examples. The full logic and the proofs of the advanced examples can be found in Appendix E.

7 Related Work and Discussions

McIver et al. [17] develop the probabilistic rely-guarantee calculus, which, to our knowledge, is the first program logic for concurrent randomized programs. Their semantics assume arbitrary schedules, i.e. the strong adversary (SA) model, and their reasonig rules use probabilistic rely/guarantee conditions. Their logic does not apply to the algorithms of conciliator and group election verified in our work, whose correctness assumes weaker adversary models. Besides, we encode probabilistic properties in the invariant and use only non-probabilistic rely-guarantee conditions, which enable simple stability proofs.

Tassarotti and Harper [19] extend the concurrent program logic Iris [15] with probabilistic relational reasoning, to establish refinements between concurrent randomized programs and monadic models. They also give rules for reasoning about probabilistic properties on monadic models. On the one hand, their program semantics assumes the SA model. On the other hand, their logic soundness only holds for schedules under which the program is guaranteed to certainly terminate (i.e. terminate in a finite number of steps). As a result, they cannot verify the examples in our work.

Fesefeldt et al. [13] propose a concurrent quantitative separation logic for reasoning about lower-bound probabilities of realizing a postcondition of a concurrent randomized program in the SA model. Like us, they require program executions to preserve invariants on shared states. But their invariants are limited to *qualitative* expectations, which map states to either 0 or 1, so cannot specify probabilistic distributions as ours. Moreover, they can only verify lower bounds of probabilities, while we can verify exact probabilities and expectations.

For the part of *sequential* reasoning, our rules mostly follow Barthe et al. [6]. Our **lclosed** condition (see the (PAR) rule in Fig. 15) is similar to their "t-closed" condition, both introduced for supporting almost surely terminating programs. Our assertion language for invariants and pre- and post-conditions is similar to theirs too, where an assertion is a predicate over state distributions. They

provide a (SPLIT) rule which is very different from our split mechanism. Using their (SPLIT) rule, one can logically split the initial distribution into two parts, reason about the execution of the same code on the two parts separately, and mix the two final distributions back. Our (SQ-OPLUS) rule for sequential reasoning in Appendix E, is almost the same as their (SPLIT) rule. It is interesting to extend our assertion language with separating conjunctions, to specify spatial disjointness of state distributions and probabilistic independence (following [7]). There are also (sequential) program logics (e.g. [9,8,1]) where assertions denote functions from program states to probabilities or expected values.

Bertrand et al. [10,11] apply model checking techniques for verifying randomized algorithms in weak adversary models. However, Bertrand et al.'s approach does not apply to the algorithms we have verified. Their work focuses on the class of algorithms with some form of "symmetry" regarding the local control flow. Such an algorithm must execute "symmetric" code for different outcomes of a coin flip. But none of the algorithms verified here satisfies this property. Instead they all have probabilistic branch statements that take different numbers of steps, which is the main challenge to our logic design. We conjecture that our split idea may still be helpful when developing automata-based approaches to verify these algorithms.

Verification overhead and scalability. One may be concerned about the verification overhead caused by adding auxiliary variables and auxiliary code, and the scalability of our logic to large algorithms. In our proofs, auxiliary variables and code are introduced to capture the key intuition of the probabilistic properties that we care about, so they are usually highly related to the random variables and the probabilistic operations (coin flips) in the original algorithms. As a result, the overhead of the auxiliary variables and code is usually proportional to the number of random variables and probabilistic operations rather than the number of lines of code. For large-scale randomized algorithms, the number of probabilistic operations may not be that large, thus the proof overhead of adding auxiliary variables and splits statements should be acceptable.

In our current setting, the auxiliary variables and split statements are added manually during the verification process, which requires a good understanding of the algorithm, i.e. how the algorithm works and why it is correct. We leave it as interesting future work to support automated code instrumentation and verification.

Acknowledgments. We thank anonymous referees for their suggestions and comments on earlier versions of this paper. This work is supported in part by National Natural Science Foundation of China (NSFC) under Grant No. 62232015.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Aguirre, A., Barthe, G., Hsu, J., Kaminski, B.L., Katoen, J.P., Matheja, C.: A pre-expectation calculus for probabilistic sensitivity. Proc. ACM Program. Lang. **5**(POPL) (jan 2021). https://doi.org/10.1145/3434333, https://doi.org/10.1145/3434333
2. Alistarh, D., Aspnes, J.: Sub-logarithmic test-and-set against a weak adversary. In: Proceedings of the 25th International Conference on Distributed Computing. pp. 97–109. DISC'11, Springer-Verlag, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24100-0_7
3. Aspnes, J.: Randomized protocols for asynchronous consensus. Distributed Comput. **16**(2-3), 165–175 (2003). https://doi.org/10.1007/s00446-002-0081-5, https://doi.org/10.1007/s00446-002-0081-5
4. Aspnes, J.: Notes on randomized algorithms (2023), https://www.cs.yale.edu/homes/aspnes/classes/469/notes.pdf
5. Aspnes, J.: Notes on theory of distributed systems (2023), https://www.cs.yale.edu/homes/aspnes/classes/465/notes.pdf
6. Barthe, G., Espitau, T., Gaboardi, M., Grégoire, B., Hsu, J., Strub, P.: An assertion-based program logic for probabilistic programs. In: Proceedings of the 27th European Symposium on Programming (ESOP 2018). pp. 117–144. Springer (2018). https://doi.org/10.1007/978-3-319-89884-1_5, https://doi.org/10.1007/978-3-319-89884-1_5
7. Barthe, G., Hsu, J., Liao, K.: A probabilistic separation logic. Proc. ACM Program. Lang. **4**(POPL), 55:1–55:30 (2020)
8. Batz, K., Kaminski, B.L., Katoen, J.P., Matheja, C.: Relatively complete verification of probabilistic programs: An expressive language for expectation-based reasoning. Proc. ACM Program. Lang. **5**(POPL) (jan 2021). https://doi.org/10.1145/3434320, https://doi.org/10.1145/3434320
9. Batz, K., Kaminski, B.L., Katoen, J.P., Matheja, C., Noll, T.: Quantitative separation logic: A logic for reasoning about probabilistic pointer programs. Proc. ACM Program. Lang. **3**(POPL) (jan 2019). https://doi.org/10.1145/3290347, https://doi.org/10.1145/3290347
10. Bertrand, N., Konnov, I., Lazic, M., Widder, J.: Verification of randomized consensus algorithms under round-rigid adversaries. In: Fokkink, W.J., van Glabbeek, R. (eds.) Proceedings of the 30th International Conference on Concurrency Theory (CONCUR 2019). LIPIcs, vol. 140, pp. 33:1–33:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019). https://doi.org/10.4230/LIPIcs.CONCUR.2019.33, https://doi.org/10.4230/LIPIcs.CONCUR.2019.33
11. Bertrand, N., Lazic, M., Widder, J.: A reduction theorem for randomized distributed algorithms under weak adversaries. In: Henglein, F., Shoham, S., Vizel, Y. (eds.) Proceedings of the 22nd International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI 2021). Lecture Notes in Computer Science, vol. 12597, pp. 219–239. Springer (2021). https://doi.org/10.1007/978-3-030-67067-2_11, https://doi.org/10.1007/978-3-030-67067-2_11
12. Chor, B., Israeli, A., Li, M.: Wait-free consensus using asynchronous hardware. SIAM Journal on Computing **23**(4), 701–712 (1994). https://doi.org/10.1137/S0097539790192635, https://doi.org/10.1137/S0097539790192635
13. Fesefeldt, I., Katoen, J., Noll, T.: Towards concurrent quantitative separation logic. In: Proceedings of 33rd International Conference on Concurrency Theory (CONCUR 2022). pp. 25:1–25:24 (2022). https://doi.org/10.4230/LIPIcs.CONCUR.2022.25, https://doi.org/10.4230/LIPIcs.CONCUR.2022.25

14. Jones, C.B.: Tentative steps toward a development method for interfering programs. ACM Trans. Program. Lang. Syst. **5**(4), 596–619 (oct 1983). https://doi.org/10.1145/69575.69577, https://doi.org/10.1145/69575.69577
15. Jung, R., Swasey, D., Sieczkowski, F., Svendsen, K., Turon, A., Birkedal, L., Dreyer, D.: Iris: Monoids and invariants as an orthogonal basis for concurrent reasoning. In: Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. p. 637–650. POPL '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2676726.2676980, https://doi.org/10.1145/2676726.2676980
16. McIver, A., Morgan, C.: Abstraction, Refinement and Proof for Probabilistic Systems. Monographs in Computer Science, Springer (2005). https://doi.org/10.1007/b138392, https://doi.org/10.1007/b138392
17. McIver, A., Rabehaja, T.M., Struth, G.: Probabilistic rely-guarantee calculus. Theor. Comput. Sci. **655**, 120–134 (2016). https://doi.org/10.1016/j.tcs.2016.01.016, https://doi.org/10.1016/j.tcs.2016.01.016
18. Rand, R., Zdancewic, S.: VPHL: A verified partial-correctness logic for probabilistic programs. In: Ghica, D.R. (ed.) Proceedings of the 31st Conference on the Mathematical Foundations of Programming Semantics (MFPS 2015). Electronic Notes in Theoretical Computer Science, vol. 319, pp. 351–367. Elsevier (2015). https://doi.org/10.1016/j.entcs.2015.12.021, https://doi.org/10.1016/j.entcs.2015.12.021
19. Tassarotti, J., Harper, R.: A separation logic for concurrent randomized programs. Proc. ACM Program. Lang. **3**(POPL) (jan 2019). https://doi.org/10.1145/3290377, https://doi.org/10.1145/3290377

A More Preliminaries

Definition 15 (finite series of real numbers). Let $r_0, \ldots, r_n$ be a finite sequence of real numbers, the finite series $\sum_{i=0}^{n} r_i$ is inductively defined as

$$\begin{array}{l} \sum_{i=0}^{0} r_i \stackrel{\text{def}}{=} 0 \\ \sum_{i=0}^{n+1} r_i \stackrel{\text{def}}{=} (\sum_{i=0}^{n} r_i) + r_{n+1}. \end{array}$$

Definition 16 (infinite series of real numbers). Let $(r_n)_{n\in\mathbb{N}}$ be an infinite sequence of real numbers, the infinite series $\sum_{i=0}^{\infty} r_i$ is defined as

$$\sum_{i=0}^{\infty} r_i \stackrel{\text{def}}{=} \lim_{n\to\infty} \sum_{i=0}^{n} r_i.$$

Definition 17 (summation on finite sets). Let A be a finite set and $f : A \to \mathbb{R}$ be a function, then there exists a bijection $g : \{0, \ldots, \#A-1\} \to A$, and $\sum_{a\in A} f(a)$ is defined as

$$\sum_{a\in A} f(a) \stackrel{\text{def}}{=} \sum_{i=0}^{\#A-1} f(g(i)).$$

We can prove the definition does not depend on the choice of g.

Definition 18 (summation on countably infinite sets). Let A be a countably infinite set and $f : A \to \mathbb{R}$ be a function, then there exists a bijection $g : \mathbb{N} \to A$. We say $\sum_{a\in A} f(a)$ is absolutely convergent iff $\sum_{i=0}^{\infty} |f(g(i))|$ converges, and the value of $\sum_{a\in A} f(a)$ is defined as

$$\sum_{a\in A} f(a) \stackrel{\text{def}}{=} \sum_{i=0}^{\infty} f(g(i)).$$

We can prove the definition does not depend on the choice of g.

Definition 19 (summation on uncountable sets). Let A be an uncountable set and $f : A \to \mathbb{R}$ be a function such that $\mathit{supp}(f) \stackrel{\text{def}}{=} \{a \in A \mid f(a) \neq 0\}$ is countable. $\sum_{a\in A} f(a)$ is defined as

$$\sum_{a\in A} f(a) \stackrel{\text{def}}{=} \sum_{a\in \mathit{supp}(f)} f(a).$$

We write $\sum_{a\in A}\{f(a) \mid P(a)\}$ as a shorthand for $\sum_{a\in\{a'\in A \mid P(a')\}} f(a)$, where P is a predicate over A.

Definition 20 ((sub-)distribution). A sub-distribution over a set A is defined by a function $\mu : A \to [0, 1]$ such that

- the support $\mathit{supp}(\mu)$ is countable; and
- the weight $|\mu| \stackrel{\text{def}}{=} \sum_{a\in A} \mu(a) \leq 1$.

If μ is a sub-distribution over A and $|\mu| = 1$, we say μ is a distribution over A. We use $\mathbb{SD}_A$ to denote the set of sub-distributions over A, and $\mathbb{D}_A$ to denote the set of distributions over A.

Definition 21 (probability of events). Let $\mu \in \mathbb{SD}_A$. The probability of an event $E : A \to \text{Prop}$ w.r.t μ is defined as

$$\mathbf{Pr}_{a\sim\mu}[E(a)] \stackrel{\text{def}}{=} \textstyle\sum_{a\in A}\{\mu(a) \mid E(a)\}$$

Definition 22 (expected value of random variables). Let $\mu \in \mathbb{SD}_A$. The expected value of a random variable $V : A \to \mathbb{R}$ w.r.t μ is defined as

$$\mathbb{E}_{a\sim\mu}[V(a)] \stackrel{\text{def}}{=} \textstyle\sum_{a\in A} \mu(a) \cdot V(a)$$

Definition 23 (expected sub-distribution). Let $\mu \in \mathbb{SD}_A$ and $f : A \to \mathbb{SD}_B$. The expected sub-distribution $\mathbb{E}_{a\sim\mu}\{f(a)\} \in \mathbb{SD}_B$ is defined as

$$\mathbb{E}_{a\sim\mu}\{f(a)\} \stackrel{\text{def}}{=} \lambda b. \textstyle\sum_{a\in A} \mu(a) \cdot f(a)(b)$$

Definition 24 (flattened sub-distribution).
Let $\mu \in \mathbb{SD}_{\mathbb{SD}_A}$. The flattened sub-distribution $\overline{\mu} \in \mathbb{SD}_A$ is defined as

$$\overline{\mu} \stackrel{\text{def}}{=} \lambda a. \textstyle\sum_{\nu\in\mathbb{SD}_A} \mu(\nu) \cdot \nu(a)\,.$$

Definition 25 (conditional sub-distribution). Let $\mu \in \mathbb{SD}_A$ and $E : A \to \text{Prop}$ such that $\mathbf{Pr}_{a\sim\mu}[E(a)] > 0$. The conditional sub-distribution $\mu|_E$ is defined as

$$\mu|_E \stackrel{\text{def}}{=} \lambda a. \begin{cases} \frac{\mu(a)}{\mathbf{Pr}_{a\sim\mu}[E(a)]}, & \text{if } E(a) \text{ holds} \\ 0, & \text{otherwise} \end{cases}$$

Definition 26 (convergent sequence of sub-distributions). Let A be a set, $\vec{\mu}$ be an infinite sequence of sub-distributions over A. We say $\vec{\mu}$ *converges to a sub-distribution* μ, represented as $\lim \vec{\mu} = \mu$, if and only if $\lim_{n\to\infty} \sum_{a\in A} |\vec{\mu}[n](a) - \mu(a)| = 0$ (where $\vec{\mu}[n]$ means the n-th element of the sequence $\vec{\mu}$). We say $\vec{\mu}$ diverges and $\lim \vec{\mu}$ is undefined if $\vec{\mu}$ does not converge to any μ.

Definition 27 (product sub-distribution). Let $\mu_1 \in \mathbb{SD}_A$ and $\mu_2 \in \mathbb{SD}_B$. The product sub-distribution $\mu_1 \otimes \mu_2 \in \mathbb{SD}_{A\times B}$ is defined as

$$\mu_1 \otimes \mu_2 \stackrel{\text{def}}{=} \lambda(a, b).\ \mu_1(a) \cdot \mu_2(b)$$

Definition 28 (projection of sub-distribution). Let $\mu \in \mathbb{SD}_{A\times B}$. and $\mu_2 \in \mathbb{SD}_B$. The projection of μ with the sets A and B is defined as:

$$\mu^{(A)} \stackrel{\text{def}}{=} \lambda a'. \mathbf{Pr}_{(a,b)\sim\mu}[a = a']$$
$$\mu^{(B)} \stackrel{\text{def}}{=} \lambda b'. \mathbf{Pr}_{(a,b)\sim\mu}[b = b']$$

B Full Operational Semantics

Fig. 17 gives full rules of concrete operational semantics. The step rules for sequencing, ifs and while-loops are mostly standard except that in the first rule for sequencing, we require that C_1 is not **skip** to prevent conflicts with the stutter rule for **skip**.

Before giving semantics to $\langle C\rangle$, we first introduce the n-step thread-local transition, represented as $(C,\sigma) \xrightarrow{p}{}^{n} (C',\sigma')$. Informally, if there is only one n-step execution path from (C,σ) to (C',σ'), the probability p in $(C,\sigma) \xrightarrow{p}{}^{n} (C',\sigma')$ is the product of the probability of every step on the path. If there are more than one execution paths, we need to sum up the probabilities of all the paths.

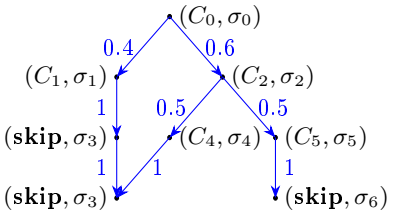


Fig. 16

Figure 16 shows an execution. There is only one 2-step path from (C_0,σ_0) to (C_5,σ_5), thus $(C_0,\sigma_0) \xrightarrow{0.6\times0.5}{}^{2} (C_5,\sigma_5)$. Similarly, there is a 2-step transition $(C_0,\sigma_0) \xrightarrow{0.4}{}^{2} (\mathbf{skip},\sigma_3)$. However, since there is another 3-step path from (C_0,σ_0) to $(\mathbf{skip},\sigma_3)$, we also have $(C_0,\sigma_0) \xrightarrow{0.7}{}^{3} (\mathbf{skip},\sigma_3)$, where the probability is the sum of $0.4\times1\times1$ and $0.6\times0.5\times1$.

Then the operational semantics rule for $\langle C\rangle$ says it finishes the execution of C in one step (that is, the execution of C cannot be interrupted by other threads). For the example in Fig. 16, we know $(\langle C_0\rangle,\sigma_0) \xrightarrow{0.7} (\mathbf{skip},\sigma_3)$ and $(\langle C_0\rangle,\sigma_0) \xrightarrow{0.3} (\mathbf{skip},\sigma_6)$. This also shows that $\langle C\rangle$ may lead to different states with different probabilities, since C may contain probabilistic choices.

The rule for atomic blocks assumes that programmers never write while loops in atomic blocks so that atomic blocks can always terminate in a bounded number of steps. We also have a more general rule for atomic blocks that permit while loops inside, as shown below. When $\langle C\rangle$ contains no while loops, the two rules are equivalent. To simplify the presentation, we choose to present the simpler rule (that avoids using limit) in this paper.

$$\frac{\forall n.\ (C,\sigma) \xrightarrow{\vec{p}\,[n]}{}^{n} (\mathbf{skip},\sigma')}{(\langle C\rangle,\sigma) \xrightarrow{\lim \vec{p}} (\mathbf{skip},\sigma')}$$

In Sec. 4.1, we give the definition of $\models_{\text{ND}} \{P\}\mathbb{C}\{Q\}$. Now we give the definition of $\models_{\text{PR}} \{P\}\mathbb{C}\{Q\}$ in Def. 29. We use $\phi \in \mathbb{D}_{Schedule}$ to denote the distribution of oblivious schedules and we define $[\![\mathbb{C}]\!]_\phi(\mu) \stackrel{\text{def}}{=} \mathbb{E}_{\varphi\sim\phi}\{[\![\mathbb{C}]\!]_\varphi(\mu)\}$ as the final state distribution of the execution forest of $\mathbb{C}$ from the initial state distribution μ under probabilistic schedules sampled from ϕ.

Definition 29. $\models_{\text{PR}} \{P\}\mathbb{C}\{Q\}$ iff, for all μ and ϕ, if $\mu \models P$, and $|[\![\mathbb{C}]\!]_\phi(\mu)| = 1$, then $[\![\mathbb{C}]\!]_\phi(\mu) \models Q$.

As explained in Sec. 3.2, $\models_{\text{ND}} \{P\}\mathbb{C}\{Q\}$ and $\models_{\text{ND}} \{P\}\mathbb{C}\{Q\}$ are equivalent when Q is closed, as shown by Thm. 3.

Theorem 3. For all $P,\mathbb{C},Q$ such that $\mathbf{closed}(Q)$, then $\models_{\text{ND}} \{P\}\mathbb{C}\{Q\} \iff \models_{\text{PR}} \{P\}\mathbb{C}\{Q\}$.

Thread IDs, schedules, states and states distributions:

$(\mathit{ThreadId})\ t \in \mathbb{N}_+ \qquad (\mathit{Schedule})\ \varphi ::= t :: \varphi \qquad$ (coinductive)

$(\mathit{State})\ \sigma \in \mathit{PVar} \to \mathbb{R} \qquad (\mathit{DState})\ \mu \in \mathbb{D}_{\mathit{State}}$

Global transitions: $(\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')$

$$\frac{(C_t, \sigma) \xrightarrow{p} (C_t', \sigma')}{(C_1 \parallel \cdots \parallel C_t \parallel \cdots \parallel C_n, \sigma) \xrightarrow[t]{p} (C_1 \parallel \cdots \parallel C_t' \parallel \cdots \parallel C_n, \sigma')}$$

Global multistep transitions: $(\mathbb{C}, \sigma) \xrightarrow{p}{}^{n} (\mathbb{C}', \sigma')$

$$\frac{}{(\mathbb{C}, \sigma) \xrightarrow[\varphi]{1}{}^{0} (\mathbb{C}, \sigma)} \qquad \frac{p = \sum_{\mathbb{C}', \sigma} \{p_1 \cdot p_2 \mid^{\varphi} (\mathbb{C}, \sigma) \xrightarrow[t]{p_1} (\mathbb{C}', \sigma') \wedge (\mathbb{C}', \sigma') \xrightarrow[\varphi]{p_2}{}^{n} (\mathbb{C}'', \sigma'')\}}{(\mathbb{C}, \sigma) \xrightarrow[t::\varphi]{p}{}^{n+1} (\mathbb{C}'', \sigma'')}$$

Thread-local transitions: $(C, \sigma) \xrightarrow{p} (C', \sigma')$

$$\frac{}{(\mathbf{skip}, \sigma) \xrightarrow{1} (\mathbf{skip}, \sigma)} \qquad \frac{[\![e]\!]_\sigma = n}{(x := e, \sigma) \xrightarrow{1} (\mathbf{skip}, \sigma\{x \rightsquigarrow n\})}$$

$$\frac{C_1 \neq \mathbf{skip} \quad (C_1, \sigma) \xrightarrow{p} (C_1', \sigma')}{(C_1; C_2, \sigma) \xrightarrow{p} (C_1'; C_2, \sigma')} \qquad \frac{}{(\mathbf{skip}; C_2, \sigma) \xrightarrow{1} (C_2, \sigma)}$$

$$\frac{[\![b]\!]_\sigma = \mathrm{tt}}{(\mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2, \sigma) \xrightarrow{1} (C_1, \sigma)} \qquad \frac{[\![b]\!]_\sigma = \mathrm{ff}}{(\mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2, \sigma) \xrightarrow{1} (C_2, \sigma)}$$

$$\frac{[\![b]\!]_\sigma = \mathrm{tt}}{(\mathbf{while}\ (b)\ \mathbf{do}\ C, \sigma) \xrightarrow{1} (C; \mathbf{while}\ (b)\ \mathbf{do}\ C, \sigma)} \qquad \frac{[\![b]\!]_\sigma = \mathrm{ff}}{(\mathbf{while}\ (b)\ \mathbf{do}\ C, \sigma) \xrightarrow{1} (\mathbf{skip}, \sigma)}$$

$$\frac{}{(\langle C_1 \rangle \oplus_p \langle C_2 \rangle, \sigma) \xrightarrow{p} (\langle C_1 \rangle, \sigma)} \qquad \frac{}{(\langle C_1 \rangle \oplus_p \langle C_2 \rangle, \sigma) \xrightarrow{1-p} (\langle C_2 \rangle, \sigma)}$$

$$\frac{\exists k. \forall n \geq k.\ (C, \sigma) \xrightarrow{p}{}^{n} (\mathbf{skip}, \sigma')}{(\langle C \rangle, \sigma) \xrightarrow{p} (\mathbf{skip}, \sigma')} \qquad \frac{(\langle C \rangle, \sigma) \xrightarrow{p} (\mathbf{skip}, \sigma')}{(\langle C \rangle\ \mathbf{split}(b_1, \ldots, b_k), \sigma) \xrightarrow{p} (\mathbf{skip}, \sigma')}$$

$$\frac{}{(C, \sigma) \xrightarrow{1}{}^{0} (C, \sigma)} \qquad \frac{p = \sum_{C', \sigma'} \{p_1 \cdot p_2 \mid (C, \sigma) \xrightarrow{p_1} (C', \sigma') \wedge (C', \sigma') \xrightarrow{p_2}{}^{n} (C'', \sigma'')\}}{(C, \sigma) \xrightarrow{p}{}^{n+1} (C'', \sigma'')}$$

Fig. 17: Full Rules for Concrete Operational Semantics

Proof. For all $P, \mathbb{C}, Q$ such that $\mathbf{closed}(Q)$, first we prove $\models_{\text{ND}} \{P\}\mathbb{C}\{Q\} \implies \models_{\text{PR}} \{P\}\mathbb{C}\{Q\}$.

To prove $\models_{\text{PR}} \{P\}\mathbb{C}\{Q\}$, we need to prove for all μ and ϕ, if $\mu \models P$ and $|[\![\mathbb{C}]\!]_\phi(\mu)| = 1$, then $[\![\mathbb{C}]\!]_\phi(\mu) \models Q$. From $1 = |[\![\mathbb{C}]\!]_\phi(\mu)| = |\mathbb{E}_{\varphi\sim\phi}\{[\![\mathbb{C}]\!]_\varphi(\mu)\}| = \mathbb{E}_{\varphi\sim\phi}[|[\![\mathbb{C}]\!]_\varphi(\mu)|] = \sum_\varphi \phi(\varphi) \cdot |[\![\mathbb{C}]\!]_\varphi(\mu)| \leq \sum_\varphi \phi(\varphi) = 1$ we know $\sum_\varphi \phi(\varphi) \cdot |[\![\mathbb{C}]\!]_\varphi(\mu)| = \sum_\varphi \phi(\varphi)$, so $\sum_\varphi \phi(\varphi)\cdot(1-|[\![\mathbb{C}]\!]_\varphi(\mu)|) = 0$, thus $\phi(\varphi)\cdot(1-|[\![\mathbb{C}]\!]_\varphi(\mu)|) = 0$ for all φ. Therefore $|[\![\mathbb{C}]\!]_\varphi(\mu)| = 1$ for all $\varphi \in \mathit{supp}(\phi)$. From $\models_{\text{ND}} \{P\}\mathbb{C}\{Q\}$ and $\mu \models P$ we know $[\![\mathbb{C}]\!]_\varphi(\mu) \models Q$ for all $\varphi \in \mathit{supp}(\phi)$. From $[\![\mathbb{C}]\!]_\phi(\mu) = \mathbb{E}_{\varphi\sim\phi}\{[\![\mathbb{C}]\!]_\varphi(\mu)\}$ and $\mathbf{closed}(Q)$ we have $[\![\mathbb{C}]\!]_\phi(\mu) \models Q$.

Next we prove $\models_{\text{PR}} \{P\}\mathbb{C}\{Q\} \implies \models_{\text{ND}} \{P\}\mathbb{C}\{Q\}$.

To prove $\models_{\text{ND}} \{P\}\mathbb{C}\{Q\}$, we need to prove for all μ and φ, if $\mu \models P$ and $|[\![\mathbb{C}]\!]_\varphi(\mu)| = 1$, then $[\![\mathbb{C}]\!]_\varphi(\mu) \models Q$. Let $\phi \stackrel{\text{def}}{=} \delta(\varphi)$, then $[\![\mathbb{C}]\!]_\phi(\mu) = \mathbb{E}_{\varphi'\sim\delta(\varphi)}\{[\![\mathbb{C}]\!]_{\varphi'}(\mu)\} = [\![\mathbb{C}]\!]_\varphi(\mu)$, so $|[\![\mathbb{C}]\!]_\phi(\mu)| = |[\![\mathbb{C}]\!]_\varphi(\mu)| = 1$. From $\models_{\text{PR}} \{P\}\mathbb{C}\{Q\}$ and $\mu \models P$ we have $[\![\mathbb{C}]\!]_\phi(\mu) \models Q$. From $[\![\mathbb{C}]\!]_\phi(\mu) = [\![\mathbb{C}]\!]_\varphi(\mu)$ we have $[\![\mathbb{C}]\!]_\varphi(\mu) \models Q$.

In Sec. 4.2, we use $\mathbf{History}(W, \varphi, \vec{W})$ to represent that $\vec{W}$ is an infinite sequence $W_0, W_1, \ldots$ where $W_i \overset{\varphi[i]}{\hookrightarrow} W_{i+1}$ for every i, as defined coinductively below (see the definition of W and $W \overset{t}{\hookrightarrow} W'$ in Fig. 10).

$$\frac{W \overset{t}{\hookrightarrow} W' \quad \mathbf{History}(W', \varphi, \vec{W}')}{\mathbf{History}(W, t :: \varphi, W :: \vec{W}')}$$

C More Discussions

C.1 Verification Overhead and Scalability

One may be concerned about the verification overhead caused by instrumenting auxiliary variables and auxiliary code and the scalability of our logic to large algorithms. In our proofs, auxiliary variables and auxiliary code are introduced to capture the key intuition of the probabilistic properties that we care about, so they are usually highly related to the random variables and the probabilistic operations (coin flips) in the original algorithms. As a result, the overhead of the auxiliary variables and auxiliary code is usually proportional to the number of random variables and probabilistic operations in the original algorithm rather than the total size of the algorithm. For large-scale randomized algorithms, the number of probabilistic operations may not be that large, thus the overhead of instrumenting auxiliary variables and splits statements may still be acceptable.

C.2 Expertise Needed to Use Our Logic

The instrumentation of auxiliary variables as well as split statements indeed requires experts' intuition on the correctness of the algorithm being verified. One should have a good understanding of the algorithm (i.e. how the algorithm

works and why it is correct). Then, by capturing the intuition and following the informal reasoning of the correctness of the algorithm, we believe it will not be difficult to appropriately instrument the auxiliary variables and split statements.

C.3 Why Split is an Explicit Operation Instead of a Logical One

The explicit instrumentation of split can simplify the logic soundness proof. Instead, if split is treated as a logical step, the formulation of the judgment semantics would involve alternating universal and existential quantifications (where the existential quantification says there exists a logical split), which would make the soundness proof much harder. The explicit split can also make the correctness proofs of programs easier to read, since the instrumented code explicitly shows where the key proof steps occur.

C.4 Usefulness of Split Beyond the OA Model

The split mechanism and closed assertions actually give us a general abstraction mechanism for compositional reasoning of randomized algorithms. Just like loop invariants abstract away the number of loops executed, and environment invariants and rely/guarantee conditions abstract away the concrete interleaving between threads, our split mechanism and closed assertions allow us to abstract away the probabilistic weights of different branches (including loops) taken in randomized algorithms. Although the mechanism is particularly useful for the OA model, it also provides insights to simplify reasoning in randomized algorithms in general.

C.5 Comparison with Operational Semantics Based Verification

Although it is possible to do verification based on concrete operational semantics directly (by inductive reasoning about execution traces or by exploring the whole state space), we prefer the logic-based approach for the following reasons.

First, our program logic supports syntax-directed verification with high-level abstraction. Users do not need to enumerate the possible executions of programs, and the detailed low-level operational semantics are hidden by the logic soundness proofs. In particular, users do not need to enumerate the schedules of the threads. This is a great advantage because an almost surely terminating program (like the algorithms we have verified) can have an infinite number of schedules and have schedules of infinite lengths. By contrast, to carry out the proofs using the operational semantics directly, users would have to consider all possible schedules, and may need co-induction over schedules, which would lead to much more complicated proofs.

Second, the program logic enables users to formalize the key intuition of algorithms. We believe that, when trying to design or understand an algorithm, people actually conduct compositional (though informal) reasoning instead of exploring the whole state space of the execution. Our logic gives a systematic way

to describe the intuition and explain the correctness through the formalization of program specifications (e.g. invariants, rely and guarantee conditions) and the insertion of the auxiliary **split**. For instance, in our informal understanding of the shared 3-sided dice example (Sec. 3.4) and the conciliator (Sec. 6), we already implicitly partition the whole state distributions. Our work identifies this implicit but crucial step in the intuition, finds a way (called **split**) to allow users to specify this step, and proves that this step is sound.

C.6 Justifications for Non-probabilistic Rely/Guarantee Conditions

As explained in Sec. 3.3, we use R, G, I to specify the interference between the current thread and its environment, where I is the probabilistic layer invariant (over state distributions) and R, G are non-probabilistic rely/guarantee conditions (over state transitions). One might suggest to replace the layer invariant by probabilistic rely/guarantee conditions to uniform probabilistic rely/guarantee conditions and non-probabilistic ones. To do that, we need to use higher-order rely/guarantee conditions $\mathcal{R}, \mathcal{G}$, but it is not obvious what type $\mathcal{R}, \mathcal{G}$ should be defined as.

One solution is to define $\mathcal{R}, \mathcal{G}$ as predicates over transitions between state distributions, which has been tried in earlier versions of our work. It indeed works in proving probabilistic properties of randomized programs. However, when we try to prove some non-probabilistic properties, such $\mathcal{R}, \mathcal{G}$ seems not to be expressive enough to support traditional rely-guarantee reasoning. For example, we might define G as $(x = 0 \ltimes y = 1) \vee (x \neq 0 \ltimes y = 2)$ for a thread with the code $\langle \textbf{if } (b) \textbf{ then } y := 1 \textbf{ else } y := 2 \rangle$ in traditional rely-gurantee reaoning. Here $\mathbf{p} \ltimes \mathbf{q}$ means the initial state of the transition satisfies $\mathbf{p}$ and the resulting state satisfies $\mathbf{q}$. In the probabilistic setting, the initial state distribution may contain some states where $x = 0$ and some states where $x \neq 0$. We want to define some $\mathcal{G}$ to express that for the initial states where $x = 0$, the corresponding resulting states satisfies $y = 1$, and for the initial states where $x \neq 0$, the correponding resulting states satisfies $y = 2$. However, this property cannot be expressed by any predicate over transitions between state distributions because we have no information about the correpondence between the initial states and the resulting states given the initial state distribution and the resulting state distribution. One may define $\mathcal{G}$ as $(\lceil x = 0 \rceil \ltimes \lceil y = 1 \rceil) \vee (\lceil x \neq 0 \rceil \ltimes \lceil y = 2 \rceil)$ or $\lceil x = 0 \vee x \neq 0 \rceil \ltimes \lceil y = 1 \vee y = 2 \rceil$. But Both of them are different from the property we want to express. What we want to express is actually $\lceil (x = 0 \ltimes y = 1) \vee (x \neq 0 \ltimes y = 2) \rceil$, which is not a predicate over transitions between state distributions.

Another solution is to define $\mathcal{R}, \mathcal{G}$ as predicates over transitions from states to state distributions. In this way $\mathcal{R}, \mathcal{G}$ are expressive enough to specify state transitions as in the traditional rely-gurantee reasoning. We can define $\mathcal{G}$ as $(x = 0 \ltimes \lceil y = 1 \rceil) \vee (x \neq 0 \ltimes \lceil y = 2 \rceil)$ for a thread with the code $\langle \textbf{if } (b) \textbf{ then } y := 1 \textbf{ else } y := 2 \rangle$, which solves the problem of the previous solution. However, such $\mathcal{R}, \mathcal{G}$ are not expressive enough to specify the initial state distribution, which makes it more difficult to prove that an assertion P is stable with respect to $\mathcal{R}$ than the previous solution. For instance, we want to prove that an assertion

$P \stackrel{\text{def}}{=} \mathbb{E}(x) = 1$ is stable under the interference of an environment thread with the code $\langle x := 1 \oplus_{0.5} \mathbf{skip}\rangle; \mathbf{skip}$ in the oblivious adversary model. It is obvious that both $\langle x := 1 \oplus_{0.5} \mathbf{skip}\rangle$ and $\mathbf{skip}$ preserves P. In the previous solution, we can define $\mathcal{R}$ as $P \ltimes P$. It is trivial to prove P is stable with respect to $\mathcal{R}$. In the current solution, one might define $\mathcal{R}$ as $\exists N.\, x = N \ltimes ((\lceil x = 1\rceil \oplus_{0.5} \lceil x = N\rceil) \vee \lceil x = N\rceil)$, where N is a logical variable used to record the initial value of x. We can see that $\mathcal{R}$ actually allows that some of the initial states execute $\langle x := 1 \oplus_{0.5} \mathbf{skip}\rangle$ while some of the initial states exeucte $\mathbf{skip}$, which will not happen in the oblivious adversary model. This makes P not stable with respect to $\mathcal{R}$. To ensure that P is stable, we need to strengthen $\mathcal{R}$ to reject the possibility that some of the initial states execute $\langle x := 1 \oplus_{0.5} \mathbf{skip}\rangle$ while some of the initial states exeucte $\mathbf{skip}$. To do that, we might need to use program counters in P and $\mathcal{R}$, which makes the proof more complicated.

The third solution is to define $\mathcal{R}, \mathcal{G}$ as predicates over distributions of state transitions, i.e., $\mathcal{R}, \mathcal{G} \in \mathbb{D}_{State \times State} \to \text{Prop}$. This solution solves the problems of the first solution and the second solution. First, $\mathcal{R}, \mathcal{G}$ are expressive enough to specify state transitions as in the traditional rely-gurantee reasoning becase we know the correspondence between the initial states and the resulting states given the distribution of state transitions. Second, $\mathcal{R}, \mathcal{G}$ are expressive enough to specify the initial state distribution becase we know the initial state distribution given the distribution of state transitions. However, $\mathcal{R}, \mathcal{G}$ cannot be used to specify the split operation. Recall that the split operation devides the current state distribution into smaller distributions and then select one of them nondeterministically. It is not obvious how to view a split operation as a distribution of state transitions. A possible solution is to extend *State* with a bottom state $\perp$ and view a split operation as a process where some of the initial states take identity transitions while other initial states move to $\perp$. This seems to be a feasible solution but we might need to be careful with the bottom state in the design of the program logic. What's more, the assertion language for $\mathcal{R}, \mathcal{G} \in \mathbb{D}_{State \times State} \to \text{Prop}$ might be very complicated.

To be brief, we want the rely/guarantee conditions to satisfy the following requirements: (1) They should be expressive enough to specify state transitions as in the traditional rely-gurantee reasoning. (2) They should be expressive enough to specify the inital state distribution to make stability easy to prove. (3) The should be expressive enough to specify the split operation. (4) The assertion language should be simple enough.

As explained in Sec. 3.3, our solution is to separate the rely/guarantee conditions for concurrent randomized programs into three components: R, G, I where I is the probabilistic layer invariant (over state distributions) and R, G are non-probabilistic rely/guarantee conditions (over state transitions). We can see that this solution satisfies the four requirements. First, R, G are expressive enough to specify state transitions as in the classical rely-guarantee reasoning. Actually R, G are exactly the classical rely/guarantee conditions. Second, I is the probabilistic layer invariant so it is expressive enough to specify the initial state distributions. Third, we can specify the split operation by I becase the split op-

eration is a transition between state distributions. Lastly, the assertion language for R, G is the same as the one in the classical rely-guarantee reasoning and the assertion language for I is the same as the one for pre/post-conditions of sequential randomized programs.

We have to admit that the expressive power of our rely/guarantee conditions are far from complete, but they are expressive enough to prove the examples in this paper. How to define complete rely/guarantee conditions is beyond the scope of our work.

C.7 Limitations and Future Directions

First, since we assume closed postconditions, non-closed properties such as probabilistic independence and (co)variance cannot be proved using our logic. It is interesting and nontrivial to explore how to verify algorithms with these properties. Second, we have not implemented or mechanized our logic. It would also be interesting to automate the code instrumentation (split) and the side-condition checks (closed, l-closed, stability). Last but not least, We will test the applicability of our rules to more real-world algorithms and keep improving our logic in practice.

D Full Assertion Language

The syntax of assertions is shown in Fig. 18, and semantics in Fig. 19. We use $\mathbf{p}$ and $\mathbf{q}$ to represent classical assertions over states, and ξ for *probabilistic expressions* such as the expected value of an arithmetic expression or the probability of a classical assertion. The expression ξ evaluates to a real number under the state distribution μ, represented as $[\![\xi]\!]_\mu$. $\mathbb{E}(e)$ evaluates to the expected value of $[\![e]\!]_\sigma$ (where $\sigma \in \mathit{supp}(\mu)$). $\mathbf{Pr}(\mathbf{q})$ evaluates to the probability of $\sigma \models \mathbf{q}$ (where $\sigma \in \mathit{supp}(\mu)$). The key definitions of expected values and probability of assertions are shown in Eqn. (1).

We also use P, Q and I to denote *probabilistic assertions* over state distributions. The assertion $\lceil \mathbf{q} \rceil$ lifts the state assertion $\mathbf{q}$ to a probabilistic assertion. It says $\mathbf{q}$ holds on all states in the support of the state distribution. The assertion $P \oplus_p Q$ holds at μ, if μ is a *mixture* of two distributions μ_0 and μ_1, which are associated with probabilities p and $1-p$, and satisfy P and Q respectively. $Q_1 \oplus Q_2$ says there exists p such that $Q_1 \oplus_p Q_2$ holds. $\bigoplus Q$ holds on μ if and only if there exists a distribution of state distribution V such that μ is the flattened distribution of V and Q holds on each state distribution in the support of V (see Eqn. (8) for the definition of flattened distribution). $\forall X.Q$ holds on μ if and only if Q holds on $\mu\{X \rightsquigarrow r\}$ for any real number r. Here, $\mu\{X \rightsquigarrow r\}$ changes the value of X to r in all states in μ. Note that X must be a logical variable. Throughout this paper, we use capital letters X to indicate that X is a logical variable and lowercase letters x to indicate that x is a program variable.

Note that $Q \oplus Q \Rightarrow Q$ may *not* hold. For instance, let's instantiate Q with $(\lceil x = 0 \rceil \vee \lceil x \neq 0 \rceil)$. A state distribution μ satisfying $Q \oplus Q$ may be a mixture of

μ_1 and μ_2 such that all the states in $supp(\mu_1)$ satisfy $x = 0$ (thus μ_1 satisfies Q) while all the states in $supp(\mu_2)$ satisfy $x \neq 0$ (thus μ_2 satisfies Q too). However, μ itself does not satisfy Q, which requires either all the states in $supp(\mu)$ satisfy $x = 0$, or all the states satisfy $x \neq 0$.

We define **true** as a syntactic sugar of $\lceil \text{true} \rceil$ which holds on all state distributions. In addition, we define $Q \mid e_1, \ldots, e_n$ to describe that Q is probabilistically independent from $e_1, \ldots, e_n$. Informally speaking, $Q \mid e_1, \ldots, e_n$ holds on μ if and only if μ can be split into multiple distributions such that each satisfies Q and the values of $e_1, \ldots, e_n$ are all deterministic.

Actions R and G are assertions over state transitions. The action $\mathbf{p} \ltimes \mathbf{q}$ means the initial state of the transition satifies $\mathbf{p}$ and the final state of the transition satifies $\mathbf{q}$. $[\mathbf{q}]$ specifies an identity transition with the states satisfying $\mathbf{q}$. $R_1 \circ R_2$ holds on (σ, σ') if and only if there exists σ'' such that R_1 holds on (σ, σ'') and R_2 holds on (σ'', σ'). It can be used to specify multistep state transitions.

$$
\begin{array}{rcl}
(\mathit{Assertion}) & \mathbf{p}, \mathbf{q} & ::= b \mid \neg \mathbf{q} \mid \mathbf{q}_1 \wedge \mathbf{q}_2 \mid \mathbf{q}_1 \vee \mathbf{q}_2 \mid \forall X. \mathbf{q} \mid \exists X. \mathbf{q} \mid \ldots \\
(\mathit{Pexp}) & \xi & ::= r \mid \mathbb{E}(e) \mid \mathbf{Pr}(\mathbf{q}) \mid \xi_1 + \xi_2 \mid \xi_1 - \xi_2 \mid \xi_1 * \xi_2 \mid \ldots \\
(\mathit{PAssertion}) & P, Q, M, I & ::= \lceil \mathbf{q} \rceil \mid \xi_1 < \xi_2 \mid \xi_1 = \xi_2 \mid \xi_1 \leq \xi_2 \mid \neg Q \mid Q_1 \wedge Q_2 \mid Q_1 \vee Q_2 \\
 & & \quad \mid \forall X. Q \mid \exists X. Q \mid Q_1 \oplus_p Q_2 \mid Q_1 \oplus Q_2 \mid \bigoplus Q \mid \ldots \\
(\mathit{Action}) & R, G & ::= \mathbf{p} \ltimes \mathbf{q} \mid [\mathbf{q}] \mid \neg R \mid R_1 \wedge R_2 \mid R_1 \vee R_2 \mid \forall X. R \mid \exists X. R \mid R_1 \circ R_2 \mid \ldots
\end{array}
$$

Fig. 18: The Assertion Language

Below we introduce the concept of semi-classical assertions.

Definition 30. An assertion Q is semi-classical, i.e., $\mathbf{scl}(Q)$ holds, if and only if, for all μ, μ', if $\mu \models Q$ and $supp(\mu') \subseteq supp(\mu)$, then $\mu' \models Q$.

We can see that semi-classical assertions only care about the support of a distribution but do not care about the probability of each state in the support. It is easy to see $\lceil \mathbf{q} \rceil$ is semi-classical, for any classical state assertion $\mathbf{q}$. Conjunction, disjunction, universal quantification and existential quantification of semi-classical assertions are also semi-classical assertions.

Semi-classical assertions are heavily used in the thread-local proofs of our examples. By separating the postconditions into probabilistic properties and non-probabilistic properties, we put all probabilistic properties into the global invariant I and keep all thread-local assertions semi-classical, which makes our thread-local proof concise and easy to understand. What's more, since semi-classical assertions do not care about probability, proving the stability of semi-classical assertions does not need complicated reasoning about probability, which makes the rely/guarantee reasoning much simpler.

In Figs. 20, 21, 22 and 23 we give sets of rules to syntactically prove properties of assertions.

Evaluation of probabilistic expressions:

$$\begin{array}{ll} [\![r]\!]_\mu \stackrel{\text{def}}{=} r & [\![\mathbb{E}(e)]\!]_\mu \stackrel{\text{def}}{=} \mathbb{E}_{\sigma\sim\mu}[[\![e]\!]_\sigma] \\ [\![\xi_1+\xi_2]\!]_\mu \stackrel{\text{def}}{=} [\![\xi_1]\!]_\mu + [\![\xi_2]\!]_\mu & [\![\mathbf{Pr}(\mathbf{q})]\!]_\mu \stackrel{\text{def}}{=} \mathbf{Pr}_{\sigma\sim\mu}[\sigma \models \mathbf{q}] \end{array}$$

Semantics of probabilistic assertions:

$\mu \models \lceil \mathbf{q} \rceil$ iff for all $\sigma \in \mathit{supp}(\mu)$, $\sigma \models \mathbf{q}$

$\mu \models Q_1 \oplus_p Q_2$ iff $p = 1$ and $\mu \models Q_1$, or $p = 0$ and $\mu \models Q_2$, or $0 < p < 1$ and there exist μ_1 and μ_2 such that $\mu = \mu_1 \oplus_p \mu_2$, $\mu_1 \models Q_1$ and $\mu_2 \models Q_2$

$\mu \models Q_1 \oplus Q_2$ iff there exists p such that $\mu \models Q_1 \oplus_p Q_2$

$\mu \models \bigoplus Q$ iff there exists $V \in \mathbb{D}_{\mathbb{D}_{State}}$ such that $\mu = \overline{V}$ and $\forall \nu \in \mathit{supp}(V).\, \nu \models Q$

$\mu\{X \rightsquigarrow r\} \stackrel{\text{def}}{=} \mathbb{E}_{\sigma\sim\mu}\{\delta(\sigma\{X \rightsquigarrow r\})\}$

$\mu \models \forall X.Q$ iff for all r, $\mu\{X \rightsquigarrow r\} \models Q$

$\mu \models \exists X.Q$ iff there exists r such that $\mu\{X \rightsquigarrow r\} \models Q$

$\mathbf{true} \stackrel{\text{def}}{=} \lceil \text{true} \rceil$ $\quad Q \mid e_1, \ldots, e_n \stackrel{\text{def}}{=} \bigoplus(\exists X_1, \ldots, X_n. \lceil e_1 = X_1 \wedge \cdots \wedge e_n = X_n \rceil \wedge Q)$

Semantics of actions:

$(\sigma, \sigma') \models \mathbf{p} \ltimes \mathbf{q}$ iff $\sigma \models \mathbf{p}$ and $\sigma' \models \mathbf{q}$

$(\sigma, \sigma') \models [\mathbf{q}]$ iff $\sigma = \sigma'$ and $\sigma \models \mathbf{q}$

$(\sigma, \sigma') \models \forall X.R$ iff for all r, $(\sigma\{X \rightsquigarrow r\}, \sigma'\{X \rightsquigarrow r\}) \models R$

$(\sigma, \sigma') \models \exists X.R$ iff there exists r such that $(\sigma\{X \rightsquigarrow r\}, \sigma'\{X \rightsquigarrow r\}) \models R$

$(\sigma, \sigma') \models R_1 \circ R_2$ iff there exists σ'' such that $(\sigma, \sigma'') \models R_1$ and $(\sigma'', \sigma') \models R_2$

$\mathbf{True} \stackrel{\text{def}}{=} \text{true} \ltimes \text{true}$ $\quad \mathbf{Id} \stackrel{\text{def}}{=} [\text{true}]$ $\quad \mathbf{Inv}(e) \stackrel{\text{def}}{=} \exists X. e = X \ltimes e = X$

$[\![R]\!] \stackrel{\text{def}}{=} \{(\sigma, \sigma') \mid (\sigma, \sigma') \models R\}$

Fig. 19: Semantics of More Assertions

Inference rules for stability of probabilistic assertions are given in Fig. 20. To prove $\lceil \mathbf{q} \rceil$ is stable with respect to R and I, it suffices to prove that $\mathbf{q}$ is stable with respect to R under the classical definition of stability. An assertion Q is stable with respect to R and Q for any R. Q is stable with respect to R and I if an equivalent assertion Q' is stable with respect to a weaker rely condition R' and a weaker invariant I'. The other rules are straightforward and need no explanation.

$$\mathbf{sta}(\mathbf{q}, R) \text{ iff } \forall \sigma, \sigma'.\ \sigma \models \mathbf{q} \wedge (\sigma, \sigma') \models R \implies \sigma' \models \mathbf{q}$$

$$\frac{\mathbf{sta}(\mathbf{q}, R)}{\mathbf{Sta}(\lceil \mathbf{q} \rceil, R, I)} \qquad \frac{\mathbf{Sta}(Q, R, I)}{\mathbf{Sta}(\forall X.Q, R, I)} \qquad \frac{\mathbf{Sta}(Q, R, I)}{\mathbf{Sta}(\exists X.Q, R, I)}$$

$$\frac{\mathbf{Sta}(Q_1, R, I) \quad \mathbf{Sta}(Q_2, R, I)}{\mathbf{Sta}(Q_1 \wedge Q_2, R, I)} \qquad \frac{\mathbf{Sta}(Q_1, R, I) \quad \mathbf{Sta}(Q_2, R, I)}{\mathbf{Sta}(Q_1 \vee Q_2, R, I)}$$

$$\frac{}{\mathbf{Sta}(Q, R, Q)} \qquad \frac{\mathbf{Sta}(Q', R', I') \quad Q \Leftrightarrow Q' \quad R \Rightarrow R' \quad I \Rightarrow I'}{\mathbf{Sta}(Q, R, I)}$$

Fig. 20: Inference Rules for Stability

Inference rules for semi-classical assertions are given in Fig. 21. It is easy to see $\lceil \mathbf{q} \rceil$ is semi-classical, for any classical state assertion $\mathbf{q}$. Conjunction, disjunction, universal quantification and existential quantification of semi-classical assertions are also semi-classical assertions.

$$\frac{}{\mathbf{scl}(\lceil \mathbf{q} \rceil)} \qquad \frac{\mathbf{scl}(Q)}{\mathbf{scl}(\forall X.Q)} \qquad \frac{\mathbf{scl}(Q)}{\mathbf{scl}(\exists X.Q)} \qquad \frac{\mathbf{scl}(Q_1) \quad \mathbf{scl}(Q_2)}{\mathbf{scl}(Q_1 \wedge Q_2)} \qquad \frac{\mathbf{scl}(Q_1) \quad \mathbf{scl}(Q_2)}{\mathbf{scl}(Q_1 \vee Q_2)}$$

Fig. 21: Inference Rules for Semi-Classical Assertions

Inference rules for closed assertions are given in Fig. 22. The lifted assertion $\lceil \mathbf{q} \rceil$ is closed. Conjunction and universal quantification of closed assertions are closed. If both Q_1 and Q_2 are closed, then $Q_1 \oplus_p Q_2$ and $Q_1 \oplus Q_2$ are closed. An assertion equivalent to a closed assertion is also closed. In addition, assertions in the form of $\xi_1 \bowtie \xi_2 \wedge Q$ where $\bowtie \in \{<, =, \leq\}$ is closed if Q is closed and Q ensures that every arithmetic expression e appears in the form of $\mathbb{E}(e)$ in ξ_1 or ξ_2 is bounded or nonnegative.

Note that disjunction, existential quantification and negation of closed assertions may *not* be closed. For instance, $\lceil x = 1 \rceil \vee \lceil x = 2 \rceil$, $\exists N. \lceil x = N \rceil$, and $\mathbf{Pr}(x = 0) \neq 0.5$ are not closed. It may be surprising to notice that, $(Q \oplus Q \Rightarrow Q)$ does not imply $\mathbf{closed}(Q)$, because the mixture of an infinite number of distributions can possibly fail Q (here we do not expand the reason which will need

$$
\begin{array}{ll}
\mathit{getExprs} & \in \mathit{Pexp} \to \mathcal{P}(\mathit{Expr}) \\
\mathit{getExprs}(r) & \stackrel{\text{def}}{=} \emptyset \\
\mathit{getExprs}(\mathbb{E}(e)) & \stackrel{\text{def}}{=} \{e\} \\
\mathit{getExprs}(\mathbf{Pr}(q)) & \stackrel{\text{def}}{=} \emptyset \\
\mathit{getExprs}(\xi_1 + \xi_2) & \stackrel{\text{def}}{=} \mathit{getExprs}(\xi_1) \cup \mathit{getExprs}(\xi_2) \\
& \vdots \\
\mathit{getExprs}(\{\xi_1, \xi_2\}) & \stackrel{\text{def}}{=} \mathit{getExprs}(\xi_1) \cup \mathit{getExprs}(\xi_2)
\end{array}
$$

$$
\frac{\mathbf{closed}(Q) \quad \forall e \in \mathit{getExprs}(\{\xi_1, \xi_2\}).\ (\exists r_1, r_2.\ Q \Rightarrow \lceil r_1 \le e \wedge e \le r_2 \rceil) \vee (Q \Rightarrow \lceil e \ge 0 \rceil)}{\mathbf{closed}(\xi_1 < \xi_2 \wedge Q)}
$$

$$
\frac{\mathbf{closed}(Q) \quad \forall e \in \mathit{getExprs}(\{\xi_1, \xi_2\}).\ (\exists r_1, r_2.\ Q \Rightarrow \lceil r_1 \le e \wedge e \le r_2 \rceil) \vee (Q \Rightarrow \lceil e \ge 0 \rceil)}{\mathbf{closed}(\xi_1 = \xi_2 \wedge Q)}
$$

$$
\frac{\mathbf{closed}(Q) \quad \forall e \in \mathit{getExprs}(\{\xi_1, \xi_2\}).\ (\exists r_1, r_2.\ Q \Rightarrow \lceil r_1 \le e \wedge e \le r_2 \rceil) \vee (Q \Rightarrow \lceil e \ge 0 \rceil)}{\mathbf{closed}(\xi_1 \le \xi_2 \wedge Q)}
$$

$$
\frac{}{\mathbf{closed}(\lceil \mathbf{q} \rceil)} \qquad \frac{\mathbf{closed}(Q_1) \quad \mathbf{closed}(Q_2)}{\mathbf{closed}(Q_1 \wedge Q_2)} \qquad \frac{\mathbf{closed}(Q)}{\mathbf{closed}(\forall X.Q)}
$$

$$
\frac{\mathbf{closed}(Q_1) \quad \mathbf{closed}(Q_2)}{\mathbf{closed}(Q_1 \oplus_p Q_2)} \qquad \frac{\mathbf{closed}(Q_1) \quad \mathbf{closed}(Q_2)}{\mathbf{closed}(Q_1 \oplus Q_2)} \qquad \frac{\mathbf{closed}(Q') \quad Q' \Leftrightarrow Q}{\mathbf{closed}(Q)}
$$

Fig. 22: Inference Rules for Closed Assertions

deep knowledge about limit). For example, $\mathbb{E}(x) = 1$ is not closed. But this does not seem to limit the applicability of our logic, because the values of program variables are often bounded, e.g. x is never less than 0, and $\mathbb{E}(x) = 1 \wedge \lceil x \ge 0 \rceil$ is closed.

In this work, we focus on the class of randomized algorithms whose correctness is about the bound of the probability of a random event or the expected value of a random variable. For thses kinds of algorithms, our syntactic rules for closedness are useful enough. The postconditions of these algorithms can usually be expressed in the form of $\Pr(b) \bowtie r$ or $\mathbb{E}(e) \bowtie r$, where r is a real number and $\bowtie$ is a comparison operator which can be $<, =, \le, >, \ge$. The closedness of the assertion $\Pr(b) \bowtie r$ can be proved directly following the syntactic rules. For $\mathbb{E}(e) \bowtie r$, we need to strengthen it with the bound of e, i.e. $\lceil r_1 \le e \le r_2 \rceil$ holds for some r_1 and r_2. Then we can prove that $(\mathbb{E}(e) \bowtie r) \wedge \lceil r_1 \le e \le r_2 \rceil$ is closed using our rules. In practice, the bounds r_1 and r_2 for e are easy to find and prove, based on the specific functionality of the verified algorithm.

Inference rules for limit-closed assertions are given in Fig. 23. The lifted assertion $\lceil \mathbf{q} \rceil$ is limit-closed. Conjunction, disjunction and universal quantification of limit-closed assertions are limit-closed. If both Q_1 and Q_2 are limit-closed, then $(Q_1 \wedge \lceil \mathbf{q} \rceil) \oplus_p (Q_2 \wedge \lceil \neg \mathbf{q} \rceil)$ and $(Q_1 \wedge \lceil \mathbf{q} \rceil) \oplus (Q_2 \wedge \lceil \neg \mathbf{q} \rceil)$ are limit-closed. In addition, assertions in the form of $\xi_1 \bowtie \xi_2 \wedge Q$ where $\bowtie \in \{=, \le\}$ is limit-closed

if Q is limit-closed and Q ensures that every arithmetic expression e appears in the form of $\mathbb{E}(e)$ in ξ_1 or ξ_2 is bounded.

$$\frac{\mathbf{lclosed}(Q) \quad \forall e \in \mathit{getExprs}(\xi_1) \cup \mathit{getExprs}(\xi_2).\ \exists r_1, r_2.\ Q \Rightarrow \lceil r_1 \le e \wedge e \le r_2 \rceil}{\mathbf{lclosed}(\xi_1 = \xi_2 \wedge Q)}$$

$$\frac{\mathbf{lclosed}(Q) \quad \forall e \in \mathit{getExprs}(\xi_1) \cup \mathit{getExprs}(\xi_2).\ \exists r_1, r_2.\ Q \Rightarrow \lceil r_1 \le e \wedge e \le r_2 \rceil}{\mathbf{lclosed}(\xi_1 \le \xi_2 \wedge Q)}$$

$$\frac{}{\mathbf{lclosed}(\lceil \mathbf{q} \rceil)} \qquad \frac{\mathbf{lclosed}(Q_1) \quad \mathbf{lclosed}(Q_2)}{\mathbf{lclosed}(Q_1 \wedge Q_2)} \qquad \frac{\mathbf{lclosed}(Q_1) \quad \mathbf{lclosed}(Q_2)}{\mathbf{lclosed}(Q_1 \vee Q_2)}$$

$$\frac{\mathbf{lclosed}(Q)}{\mathbf{lclosed}(\forall X.Q)} \qquad \frac{\mathbf{lclosed}(Q') \quad Q' \Leftrightarrow Q}{\mathbf{lclosed}(Q)}$$

$$\frac{\mathbf{lclosed}(Q_1) \quad \mathbf{lclosed}(Q_2)}{\mathbf{lclosed}((Q_1 \wedge \lceil \mathbf{q} \rceil) \oplus_p (Q_2 \wedge \lceil \neg \mathbf{q} \rceil))} \qquad \frac{\mathbf{lclosed}(Q_1) \quad \mathbf{lclosed}(Q_2)}{\mathbf{lclosed}((Q_1 \wedge \lceil \mathbf{q} \rceil) \oplus (Q_2 \wedge \lceil \neg \mathbf{q} \rceil))}$$

Fig. 23: Inference Rules for Limit-closed Assertions

E Extensions to Logic Rules

Unfortunately the rules in Fig. 15 are not sufficient for verifying more advanced examples, such as group election [2] and the multiplayer level-up game, whose loops require split in the first few rounds only. In this section, we motivate our extensions to the logic and give the full set of logic rules.

Example: a multiplayer level-up game. The program $\mathbb{C}_{LvUp}$ consists of n threads, where every thread i runs the following code $LvUp_i$ (ignore the code in red for now):

1 $k_i = 1$;
2 **while** $(k_i \le m \wedge v_i = 0)$ **do**
3 $\quad \langle x[k_i] := x[k_i] + 1; y[k_i] := y[k_i] + 1 \rangle \oplus_p \langle v_i := 1; y[k_i] := y[k_i] + 1 \rangle$;
4 $\quad k_i := k_i + 1$;

The game has m levels. In the shared array $x[1..m]$, each $x[j]$ records the number of threads that has passed the level j. At each level, the threads try to progress to the next level, succeeding with probability p. If a thread i fails at some level, it sets v_i to 1 and exits the game. Each thread i has two local variables: k_i records its next level, and v_i records whether it has failed the game.

Intuition. We want to verify that $\mathbb{C}_{LvUp}$ satisfies the postcondition $\forall j.\ \mathbb{E}(x[j]) = n \cdot p^j$. To see why this holds, we first observe the following when the program terminates:

- $\mathbb{E}(x[1]) = n \cdot p$. Clearly all the n threads can enter the first round of the loop and flip the coin at line 3. $\mathbb{E}(x[1])$ is the expected number of heads in these n flips, so it is $n \cdot p$.
- $\forall j.\ \mathbb{E}(x[j+1]) = p \cdot \mathbb{E}(x[j])$. Similarly, $x[j]$ is the number of threads that can enter the $(j+1)$-th round of the loop and flip the coin. So $\mathbb{E}(x[j+1]) = p \cdot \mathbb{E}(x[j])$.

Then, by induction, we know the postcondition holds. To turn this intuition into an invariant, we introduce an auxiliary array $y[1..m]$ with each $y[j]$ recording the number of threads that execute line 3 of round j. We instrument line 3 with the auxiliary code (in red above) that increments $y[k_i]$. So, when the program terminates, $\forall j.\ x[j] = y[j+1]$ holds. We formulate the invariant as follows:

$$I_{LvUp} \stackrel{\text{def}}{=} \forall j.\ Q_j, \qquad \text{where } Q_j \stackrel{\text{def}}{=} \mathbb{E}(x[j]) = p \cdot \mathbb{E}(y[j])$$

The key of the proof is to show that I_{LvUp} is indeed an invariant. Since I_{LvUp} is defined as $\forall j.\ Q_j$, we only need to prove, for every j, Q_j is an invariant. Unfortunately, just as in the *Dice* example, it is possible that the transitions at the same layer are made by different code, making it difficult to prove that Q_j is preserved layer by layer.

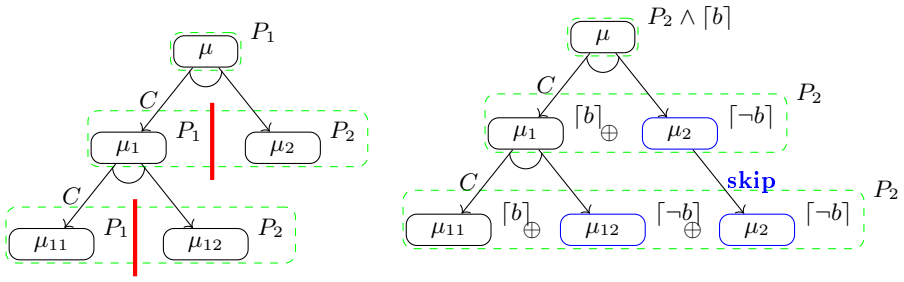
Split. Apparently line 3 of round j is the only code that modifies $x[j]$ and $y[j]$. So it is the only line that can possibly invalidate Q_j. To prove Q_j can be preserved, we split the state distributions *before* round j according to the value of v_i, in order to separate out the states at which the thread *exits* the loop and has no change to enter round j, from the states at which the thread *enters* the next round (thus has a chance to continue to round j).

After round j (i.e. $k_i > j$), the execution will not access $x[j]$ or $y[j]$, and hence will naturally preserve Q_j. In this case, we do not need to split the state distribution even though it may contain different code (code either in the loop body or **skip**), as neither of them access $x[j]$ or $y[j]$ anyway.

The challenging case is the *exact round* j (i.e. $k_i = j$), where $x[j]$ and $y[j]$ are actually updated. Suppose Q_j holds before executing line 3. The left branch of line 3 increments both $x[j]$ and $y[j]$ with probability p, while the right branch only increments $y[j]$ with probability $1-p$. We can see that Q_j would be *invalid* if we consider the distribution resulting from only one of the branches independently. It is preserved only if we consider the whole distribution resulting from both branches. Therefore, at the end of line 3 we *cannot* split the distribution based on the value of v_i, otherwise Q_j becomes invalid after the split.

As such, we insert $\mathbf{split}(k_i < j \wedge v_i = 0, k_i < j \wedge v_i = 1, k_i \geq j)$ at line 3, just after the probabilistic choice, to split the state distributions into smaller ones satisfying $\lceil v_i = 0 \rceil$ and $\lceil v_i = 1 \rceil$ when (and only when) $k_i < j$.

However, as we explain above, the third partition (where $k_i \geq j$) resulting from the split will produce distributions whose supports contain different code. Our logic rules cannot reason about this case because they always rely on proper splits of distributions to avoid the mix of code. We have to extend our logic to verify examples like *LvUp*.



(a) split in two rounds, where $P_1 \Rightarrow \lceil b \rceil$

(b) non-split, where blue boxes indicate termination of the loop's executions

Fig. 24: Execution of the loop in the (WHILE-NST) rule.

New rules. The (WHILE) rule in Fig. 15 cannot apply to the loop of *LvUp*. The reason is, the rule requires the loop invariant be $P_1 \vee P_2$ such that $P_1 \Rightarrow \lceil b \rceil$ and $P_2 \Rightarrow \lceil \neg b \rceil$, which requires one to split the state distributions into smaller ones satisfying $\lceil b \rceil$ and $\lceil \neg b \rceil$ in all rounds of the loop; but in the loop of *LvUp*, we must only split when $k_i < j$ (i.e. at the first $j-1$ rounds).

To reason about loops that split in the first few rounds only, we design a new (WHILE-NST) rule as below:

$$\frac{\begin{array}{c}\mathbf{Sta}(P_1 \vee P_2, R, I)\ P_1 \Rightarrow \lceil b \rceil\ R, G_1, I \vdash \{P_1\} C \{P_1 \vee P_2\}\ P_2 \wedge \lceil \neg b \rceil \Rightarrow Q \\ P_2 \wedge \lceil b \rceil \Rightarrow \lceil \mathbf{q} \rceil\ R, G_2, \lceil \mathbf{q} \rceil \vdash_{\text{NST}} \{P_2 \wedge \lceil b \rceil\} C \{P_2\}\ \forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x) \\ \mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C)\ \mathbf{sta}(\mathbf{q}, R)\ \mathbf{Sta}(\{P_2, Q\}, R, \mathbf{true})\ \mathbf{closed}(Q)\ \mathbf{scl}(P_2)\end{array}}{R, G_1 \vee G_2, I \vdash_{\text{NST}} \{P_1 \vee P_2\} \mathbf{while}\ (b)\ \mathbf{do}\ C \{Q\}}\ (\text{WHILE-NST})$$

Like the (WHILE) rule in Fig. 15, the (WHILE-NST) rule requires the loop invariant be a disjunction $P_1 \vee P_2$ such that $P_1 \Rightarrow \lceil b \rceil$. But it does not require $P_2 \Rightarrow \lceil \neg b \rceil$. For the loop in *LvUp*, P_1 and P_2 are instantiated as $\lceil k_i \leq j \wedge v_i = 0 \rceil$ and $\lceil k_i \leq j \wedge v_i = 1 \rceil \vee \lceil k_i > j \rceil$ respectively. With the precondition P_1, one verifies that the loop body C satisfies $R, G_1, I \vdash \{P_1\} C \{P_1 \vee P_2\}$, which requires C to properly split the state distributions to regain $P_1 \vee P_2$, just as in the (WHILE-ST) rule. With the precondition P_2, the states exiting the loop satifies Q because P_2 is semi-classical and $P_2 \wedge \lceil \neg b \rceil \Rightarrow Q$. The states entering the loop satifies $P_2 \wedge \lceil b \rceil$ due to the same reason. The (WHILE-NST) rule requires us to verify C with the precondition $P_2 \wedge \lceil b \rceil$. It requires that, starting from state distributions satisfying $P_2 \wedge \lceil b \rceil$, the execution of C satisfies the two requirements: (1) It does not modify the free variables in the invariant I. (2) It does not split the state distribution into smaller ones. In this way, the execution of C naturally perserves I.

We formulate (1) as the premise $\forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x)$. We formulate (2) as the premise $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C)$ where $\lceil \mathbf{q} \rceil$ is an invariant throughout the execution of C. $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C)$ means for any state distribution μ satisfying $\lceil \mathbf{q} \rceil$, every split instruction in C cannot split μ into smaller ones. The formal

definition of **disablesplit** is given in Definition 32. To ensure $\lceil \mathbf{q} \rceil$ is an invariant, we require that all states entering the loop satisfies $\mathbf{q}$, i.e., $P_2 \wedge \lceil b \rceil \Rightarrow \lceil \mathbf{q} \rceil$, and $\mathbf{q}$ is preserved by the environment, i.e., $\mathbf{sta}(\mathbf{q}, R)$. Note that different states may terminate the loop after executing different number of steps. However, the invariant I holds on the whole state distribution at each layer, which means conditional distribution of terminating states in each layer, which satifies Q, may not satifies I. To ensure that Q will not be invalidate by the environment, it is required that Q is stable under the interference from the environment without the assumption that the environment preserves I. That's why we need $\mathbf{Sta}(Q, R, \mathbf{true})$. $\mathbf{Sta}(P_2, R, \mathbf{true})$ is required due to the same reason. What's more, we need $\mathbf{closed}(Q)$ to conclude that the terminating state distribution satifies Q from the fact that states terminating with any number of steps satifies Q.

Note that in the (WHILE-NST) rule, we use a new kind of judgment $R, G, I \vdash_{\text{NST}} \{P\}C\{Q\}$. Here "NST" is short for "non-simultaneous termination". Roughly it means, for any state distribution μ satisfying P, the executions of C starting from a state in the support of μ can (probabilistically) terminate within different numbers of steps, and we keep all the probabilistic states upon termination in the same distribution (i.e. we do not split). This judgment is introduced for characterizing the executions of **while**-loops in which we do not split in all rounds.

Figure 24 shows executions of the different rounds of the loop. In the first few rounds, we always split (see Fig. 24a), so that the execution of the loop body C starting from P_1 terminates at separate state distributions satisfying P_1 and P_2 respectively. In this case, we verify C with $R, G_1, I \vdash \{P_1\}C\{P_1 \vee P_2\}$. For later rounds, we do not split (see Fig. 24b), so the final state distribution when the loop terminates is a mixture of μ_2 (corresponding to termination in one round), μ_{12} (corresponding to termination in two rounds), and so on. That's why in the (WHILE-NST) rule we conclude with the NST judgment for the **while**-loop. Moreover we allow each of the later rounds to non-simultaneously terminate, so we verify the NST judgment $R, G_2, I_2 \vdash_{\text{NST}} \{P_2 \wedge \lceil b \rceil\}C\{P_2\}$.

We provide a set of rules for the NST judgment. Notably, the (SEQ-NST) rule for sequential composition of the NST judgments is as below:

$$\frac{\begin{array}{c} R \vee G_2, G_1, I \vdash_{\text{NST}} \{P\}C_1\{M\}\ R, G_2, \mathbf{true} \vdash_{\text{NST}} \{M\}C_2\{Q\} \\ \forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x)\ \mathbf{Nosplit}(C_2)\ \mathbf{closed}(Q)\ \mathbf{scl}(M) \end{array}}{R, G_1 \vee G_2, I \vdash_{\text{NST}} \{P\}C_1; C_2\{Q\}}\ (\text{SEQ-NST})$$

Since the proof of C_1 uses the NST judgment, C_1 may have multiple execution traces that terminate in different numbers of steps (like the loop in Fig. 24b). Consequently, in the execution of $C_1; C_2$, different statements may be executed "at the same time". Therefore we encounter the same problem as in Sec. 3.5. However, we can borrow the ideas of the (WHILE-NST) rule and conservatively require that the execution of C_2 neither modifies the free variables in the invariant I nor splits the state distributions into smaller ones. The first requirement is formulated as the premise $\forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x)$. The second requirement is formulated as $\mathbf{Nosplit}(C_2)$, which means that C_2 contains no **split** commands.

Note that C_1 is verified under the rely condition $R \vee G_2$, because the execution of C_2 may still influence the behaviors of C_1 by modifying other variables used in C_1. What's more, we require $\mathbf{closed}(Q)$ to conclude that the terminating state distribution satifies Q from the fact that states terminating with any number of steps satifies Q. Moreover, the execution of C_1 terminates within different number of steps. To use the post-condition M as the pre-condition of C_2, we need M to be semi-classical.

Although the resulting (SEQ-NST) rule looks very restrictive, we expect that, for most examples, we can properly split at all rounds of the loops and verify them using the (WHILE) and (SEQ) rules. The (SEQ-NST) rule is supposed to be used only for advanced examples, where the thread's code is in the form of $\mathbf{while}\ (b)\ \mathbf{do}\ C; C_2$ and we split in the first few rounds of the loop only.

The full version of the extended logic rules are given in Fig. 25, Fig. 26 and Fig. 27. We use $R, G, I \vdash_{\text{ST}} \{P\}C\{Q\}$ to denote the judgement $R, G, I \vdash \{P\}C\{Q\}$ in Sec. 5.2 to highlight its difference from the judgement $R, G, I \vdash_{\text{NST}} \{P\}C\{Q\}$, where "ST" is short for "simultaneous termination" and "NST" is short for "non-simultaneous termination".

The full version of whole-program rules is given in Fig. 25. The (REMOVESPLIT) rule and the (LAZYCOIN) rule are the same as that in Fig. 15. The (PAR) rule is almost the same as the one in Fig. 15 except that we only need to prove $R_i, G_i, I \vdash_{\text{NST}} \{P_i\}C\{Q_i\}$ for every thread instead of proving $R_i, G_i, I \vdash_{\text{ST}} \{P_i\}C\{Q_i\}$ for the need of supporting more advanced examples in Sec. G. In addition, we provide the standard consequence rule (P-CSQ). The (BIGCONJ) and (BIGDISJ) rule are simple generalizations of the standard conjunction rule and disjunction rule in hoare logic.

$$\frac{P \Rightarrow P_1 \quad \vdash_{\text{A}} \{P_1\}\mathbb{C}\{Q_1\} \quad Q_1 \Rightarrow Q}{\vdash_{\text{A}} \{P\}\mathbb{C}\{Q\}}\ (\text{P-CSQ})$$

$$\frac{\vdash_{\text{A}} \{P_1\}\mathbb{C}\{Q_1\} \quad \dots \quad \vdash_{\text{A}} \{P_n\}\mathbb{C}\{Q_n\}}{\vdash_{\text{A}} \{P_1 \wedge \cdots \wedge P_n\}\mathbb{C}\{Q_1 \wedge \cdots \wedge Q_n\}}\ (\text{BIGCONJ})$$

$$\frac{\vdash_{\text{A}} \{P_1\}\mathbb{C}\{Q_1\} \quad \dots \quad \vdash_{\text{A}} \{P_n\}\mathbb{C}\{Q_n\}}{\vdash_{\text{A}} \{P_1 \vee \cdots \vee P_n\}\mathbb{C}\{Q_1 \vee \cdots \vee Q_n\}}\ (\text{BIGDISJ})$$

$$\frac{\vdash_{\text{A}} \{P\}\mathbb{C}\{Q\} \quad \mathbf{closed}(Q)}{\vdash_{\text{A}} \{P\}\mathbf{RemoveSplit}(\mathbb{C})\{Q\}}\ (\text{REMOVESPLIT}) \qquad \frac{\vdash_{\text{A}} \{P\}\mathbf{lazycoin}(\mathbb{C})\{Q\}}{\vdash_{\text{A}} \{P\}\mathbb{C}\{Q\}}\ (\text{LAZYCOIN})$$

$$\frac{\begin{array}{c}\forall i, j.\ i \neq j \implies G_j \Rightarrow R_i \qquad \forall i.\ R_i, G_i, I \vdash_{\text{NST}} \{P_i\}C_i\{Q_i\} \\ P \Rightarrow I \wedge P_1 \wedge \cdots \wedge P_n \qquad I \wedge Q_1 \wedge \cdots \wedge Q_n \Rightarrow Q \qquad \mathbf{lclosed}(\{I, Q_1, \dots, Q_n\})\end{array}}{\vdash_{\text{A}} \{P\}C_1 \parallel \cdots \parallel C_n\{Q\}}\ (\text{PAR})$$

Fig. 25: Whole Program Rules: Full Version

The full version of our thread-local rules is given in Fig. 26. The $\square$ symbol is used in many thread-local rules. It can be instantiated to ST or NST in those rules.

The (ST-NST) rule allows us to derive a $\vdash_{\text{NST}}$ judgement by deriving its $\vdash_{\text{ST}}$ counterpart. The rules for deriving $\vdash_{\text{ST}}$ judgements are easier to use than the rules for deriving $\vdash_{\text{NST}}$ judgements. If the executions of C starting from all initial states indeed terminate after the same numbers of steps, we can prove $R, G, I \vdash_{\text{NST}} \{P\}C\{Q\}$ using the (ST-NST) rule and the rules for deriving $\vdash_{\text{ST}}$ judgements.

The disjunction rule (DISJ) and the conjunction rule (CONJ) are standard. The (EXIST) rule and the (FORALL) rule introduce existential quantification and universal quantification over pre/post-conditions, respectively, as long as the quantified variable is not free in R, G, I and cannot be modified by C. The consequence rule (CSQ) allows adaptions of different parts of the judgement.

In the (SKIP) rule, the postcondition is the same as the precondition for the **skip** statement does not modify any variable. The guarantee condition is **Id** due to the same reason. In addtion, the pre/post-condition Q is required to be stable under the interference from the environment (specified using R and I), which ensures that the environment does not invalidate Q.

The (ATOM) rule and the (ATOM-SPLIT) rule are the same as that in Fig. 15. The (SEQ-ST) rule is generalized from the (SEQ) rule in Fig. 15. If $\square$ is instantiated to ST, it is the same as the (SEQ) rule. The (SEQ-ST) rule also works when $\square$ is instantiated to NST for we only require that the execution of C_1 terminate simultaneously on different states. Similarly, the (COND) rule also works when $\square$ is instantiated to NST. The (WHILE-ST) rule is the same as the one in Fig. 15.

The following definition gives the condition when a split command *sp* is disabled by an assertion Q, which is used in the (WHILE-NST) rule.

Definition 31. A split instruction $\mathbf{split}(b_1, \ldots, b_k)$ is disabled by an assertion Q, i.e.,
$\mathbf{disablesplit}(Q, \mathbf{split}(b_1, \ldots, b_k))$ holds if and only if there exists i such that $Q \Rightarrow \lceil b_i \rceil$.

Definition 32. $\mathbf{disablesplit}(Q, C)$ holds if and only if every split instruction in C is disabled by Q.

The sequential rules are given in Fig. 27. The judgement for sequential rules is in the form of $G \vdash_{\text{SQ}} \{P\}C\{Q\}$. Note that the guarantee G does not specify the state transition of every single step of C. Instead it specifies the state transitions from initial states to the corresponding final states at the end of C.

For the probabilistic choice $\langle C_1 \rangle \oplus_p \langle C_2 \rangle$, the (SQ-PCH) rule asks one to verify C_1 and C_2 separately and mix their postconditions Q_1 and Q_2 according to the probability of the coin flip. The (SQ-COND) rule takes a similar form: one only needs to verify the **then**- and **else**-branches separately.

The (SQ-SEQ) rule is similar to the standard sequential composition rule. Note that the intermediate assertion M may specify a state distribution consisting of states at different layers. The guarantee condition of $C_1; C_2$ is the composition of the guarantee condition of C_1 and the guarantee condition of C_2 for the

$$\frac{R, G, I \vdash_{\text{ST}} \{P\}C\{Q\}}{R, G, I \vdash_{\text{NST}} \{P\}C\{Q\}}\ (\text{ST-NST})$$

$$\frac{\begin{array}{c}R, G, I \vdash_{\square} \{P_1\}C\{Q_1\} \\ R, G, I \vdash_{\square} \{P_2\}C\{Q_2\}\end{array}}{R, G, I \vdash_{\square} \{P_1 \vee P_2\}C\{Q_1 \vee Q_2\}}\ (\text{DISJ})\quad \frac{\begin{array}{c}R, G, I \vdash_{\square} \{P_1\}C\{Q_1\} \\ R, G, I \vdash_{\square} \{P_2\}C\{Q_2\}\end{array}}{R, G, I \vdash_{\square} \{P_1 \wedge P_2\}C\{Q_1 \wedge Q_2\}}\ (\text{CONJ})$$

$$\frac{\begin{array}{c}X \notin fv(R, G, I) \cup wv(C) \\ R, G, I \vdash_{\square} \{P\}C\{Q\}\end{array}}{R, G, I \vdash_{\square} \{\exists X.P\}C\{\exists X.Q\}}\ (\text{EXIST})\quad \frac{\begin{array}{c}X \notin fv(R, G, I) \cup wv(C) \\ R, G, I \vdash_{\square} \{P\}C\{Q\}\end{array}}{R, G, I \vdash_{\square} \{\forall X.P\}C\{\forall X.Q\}}\ (\text{FORALL})$$

$$\frac{P \Rightarrow P_1 \quad R \Rightarrow R_1 \quad G_1 \Rightarrow G \quad Q_1 \Rightarrow Q \quad R_1, G_1, I \vdash_{\square} \{P_1\}C\{Q_1\}}{R, G, I \vdash_{\square} \{P\}C\{Q\}}\ (\text{CSQ})$$

$$\frac{\mathbf{Sta}(Q, R, I)}{R, \mathbf{Id}, I \vdash_{\text{ST}} \{Q\}\mathbf{skip}\{Q\}}\ (\text{SKIP})\quad \frac{\mathbf{Sta}(\{P, Q\}, R, I) \quad G \vdash_{\text{SQ}} \{I \wedge P\}C\{I \wedge Q\}}{R, G, I \vdash_{\text{ST}} \{P\}\langle C\rangle\{Q\}}\ (\text{ATOM})$$

$$\frac{\begin{array}{c}G \vdash_{\text{SQ}} \{I \wedge P\}C\{(I \wedge Q \wedge \lceil b_1 \rceil) \oplus \cdots \oplus (I \wedge Q \wedge \lceil b_k \rceil)\} \\ \mathbf{Sta}(\{P, Q \wedge (\lceil b_1 \rceil \vee \cdots \vee \lceil b_k \rceil)\}, R, I)\end{array}}{R, G, I \vdash \{P\}\langle C\rangle\ \mathbf{split}(b_1, \ldots, b_k)\{(Q \wedge \lceil b_1 \rceil) \vee \cdots \vee (Q \wedge \lceil b_k \rceil)\}}\ (\text{ATOM-SPLIT})$$

$$\frac{R, G, I \vdash_{\text{ST}} \{P\}C_1\{M\} \quad R, G, I \vdash_{\square} \{M\}C_2\{Q\}}{R, G, I \vdash_{\square} \{P\}C_1; C_2\{Q\}}\ (\text{SEQ-ST})$$

$$\frac{\begin{array}{c}R \vee G_2, G_1, I \vdash_{\text{NST}} \{P\}C_1\{M\}\ R, G_2, \mathbf{true} \vdash_{\text{NST}} \{M\}C_2\{Q\} \\ \forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x)\ \mathbf{Nosplit}(C_2)\ \mathbf{closed}(Q)\ \mathbf{scl}(M)\end{array}}{R, G_1 \vee G_2, I \vdash_{\text{NST}} \{P\}C_1; C_2\{Q\}}\ (\text{SEQ-NST})$$

$$\frac{\begin{array}{c}\mathbf{Sta}(P_1 \vee P_2, R, I) \qquad P_1 \Rightarrow \lceil b \rceil \qquad P_2 \Rightarrow \lceil \neg b \rceil \\ R, G, I \vdash_{\square} \{P_1\}C_1\{Q\} \qquad R, G, I \vdash_{\square} \{P_2\}C_2\{Q\}\end{array}}{R, G, I \vdash_{\square} \{P_1 \vee P_2\}\mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2\{Q\}}\ (\text{COND})$$

$$\frac{\mathbf{Sta}(\{P_1 \vee P_2, Q\}, R, I) \quad P_1 \Rightarrow \lceil b \rceil \quad P_2 \Rightarrow \lceil \neg b \rceil \wedge Q \quad R, G, I \vdash_{\text{ST}} \{P_1\}C\{P_1 \vee P_2\}}{R, G, I \vdash_{\text{ST}} \{P_1 \vee P_2\}\mathbf{while}\ (b)\ \mathbf{do}\ C\{Q\}}\ (\text{WHILE-ST})$$

$$\frac{\begin{array}{c}\mathbf{Sta}(P_1 \vee P_2, R, I)\ P_1 \Rightarrow \lceil b \rceil\ R, G_1, I \vdash_{\text{ST}} \{P_1\}C\{P_1 \vee P_2\}\ P_2 \wedge \lceil \neg b \rceil \Rightarrow Q \\ P_2 \wedge \lceil b \rceil \Rightarrow \lceil \mathbf{q} \rceil\ R, G_2, \lceil \mathbf{q} \rceil \vdash_{\text{NST}} \{P_2 \wedge \lceil b \rceil\}C\{P_2\}\ \forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x) \\ \mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C)\ \mathbf{sta}(\mathbf{q}, R)\ \mathbf{Sta}(\{P_2, Q\}, R, \mathbf{true})\ \mathbf{closed}(Q)\ \mathbf{scl}(P_2)\end{array}}{R, G_1 \vee G_2, I \vdash_{\text{NST}} \{P_1 \vee P_2\}\mathbf{while}\ (b)\ \mathbf{do}\ C\{Q\}}\ (\text{WHILE-NST})$$

Fig. 26: Thread-local Rules: Full Version

$$\frac{G \vdash_{\text{SQ}} \{P_1\}C\{Q_1\} \quad G \vdash_{\text{SQ}} \{P_2\}C\{Q_2\}}{G \vdash_{\text{SQ}} \{P_1 \vee P_2\}C\{Q_1 \vee Q_2\}}\ (\text{SQ-DISJ}) \qquad \frac{G \vdash_{\text{SQ}} \{P_1\}C\{Q_1\} \quad G \vdash_{\text{SQ}} \{P_2\}C\{Q_2\}}{G \vdash_{\text{SQ}} \{P_1 \wedge P_2\}C\{Q_1 \wedge Q_2\}}\ (\text{SQ-CONJ})$$

$$\frac{X \notin fv(G) \cup wv(C) \quad G \vdash_{\text{SQ}} \{P\}C\{Q\}}{G \vdash_{\text{SQ}} \{\exists X.P\}C\{\exists X.Q\}}\ (\text{SQ-EXIST}) \qquad \frac{X \notin fv(G) \cup wv(C) \quad G \vdash_{\text{SQ}} \{P\}C\{Q\}}{\vdash_{\text{SQ}} \{\forall X.P\}C\{\forall X.Q\}}\ (\text{SQ-FORALL})$$

$$\frac{P \Rightarrow P' \quad G' \vdash_{\text{SQ}} \{P'\}C\{Q'\} \quad Q' \Rightarrow Q \quad G' \Rightarrow G}{G \vdash_{\text{SQ}} \{P\}C\{Q\}}\ (\text{SQ-CSQ})$$

$$\frac{G \vdash_{\text{SQ}} \{P_1\}C\{Q_1\} \quad G \vdash_{\text{SQ}} \{P_2\}C\{Q_2\}}{G \vdash_{\text{SQ}} \{P_1 \oplus_p P_2\}C\{Q_1 \oplus_p Q_2\}}\ (\text{SQ-OPLUS}) \qquad \frac{G \vdash_{\text{SQ}} \{P\}C\{Q\}}{G \vdash_{\text{SQ}} \{\bigoplus P\}C\{\bigoplus Q\}}\ (\text{SQ-BIGOPLUS})$$

$$\frac{}{\mathbf{Id} \vdash_{\text{SQ}} \{Q\}\mathbf{skip}\{Q\}}\ (\text{SQ-SKIP}) \qquad \frac{G_1 \vdash_{\text{SQ}} \{P\}C_1\{M\} \quad G_2 \vdash_{\text{SQ}} \{M\}C_2\{Q\}}{G_1 \circ G_2 \vdash_{\text{SQ}} \{P\}C_1; C_2\{Q\}}\ (\text{SQ-SEQ})$$

$$\frac{P \Rightarrow Q[e/x] \quad \forall \mu, \sigma.\ \mu \models P \wedge \sigma \in \mathit{supp}(\mu) \implies (\sigma, \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}) \models G}{G \vdash_{\text{SQ}} \{P\}x := e\{Q\}}\ (\text{SQ-ASGN})$$

$$\frac{G \vdash_{\text{SQ}} \{P_1 \wedge \lceil b \rceil\}C_1\{Q_1\} \quad G \vdash_{\text{SQ}} \{P_2 \wedge \lceil \neg b \rceil\}C_2\{Q_2\}}{G \vdash_{\text{SQ}} \{(P_1 \wedge \lceil b \rceil) \oplus_p (P_2 \wedge \lceil \neg b \rceil)\}\mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2\{Q_1 \oplus_p Q_2\}}\ (\text{SQ-COND})$$

$$\frac{G \vdash_{\text{SQ}} \{P\}C\{Q\}}{G \vdash_{\text{SQ}} \{P\}\langle C\rangle\{Q\}}\ (\text{SQ-ATOM}) \qquad \frac{G \vdash_{\text{SQ}} \{P\}C_1\{Q_1\} \quad G \vdash_{\text{SQ}} \{P\}C_2\{Q_2\}}{G \vdash_{\text{SQ}} \{P\}\langle C_1\rangle \oplus_p \langle C_2\rangle\{Q_1 \oplus_p Q_2\}}\ (\text{SQ-PCH})$$

Fig. 27: Sequential Rules: Full Version

guarantee condition in the sequential judgement specifies the transition from the initial state to the terminating state. For example, we can prove

$$(x = y \ltimes x = y) \vdash_{\text{SQ}} \{\lceil x = y \rceil\}x := x + 1; y := y + 1\{\lceil x = y \rceil\}$$

using the (SQ-SEQ) rule by instantiating G_1 with $(x = y \ltimes x = y + 1)$ and G_2 with $(x = y + 1 \ltimes x = y)$.

The (SQ-OPLUS) rule and the (SQ-BIGOPLUS) rule are useful for local reasoning and reflect the additivity of the semantics. Using the (SQ-OPLUS) rule, we can logically split the initial state distribution into two parts, reasoning about the execution of C on the two parts separately, and mix the two terminating state distributions back according to their weight in the initial state distribution. The (SQ-BIGOPLUS) rule is similar but allows us to logically split the initial state distribution into infinite ones. The other sequential rules are direct generalizations of the classical Hoare logic rules to the probabilistic setting.

F Judgement Semantics

The semantics of the top-level judgement $\vdash_A \{P\}\mathbb{C}\{Q\}$ has already been defined in Definition 12 based on the abstract operational semantics.

Before giving the definition of the thread-local judgement, we define thread-local transitions in Fig. 28. As in the abstract operational semantics, we model the execution of a thread under the interference from the environment as transitions between the sub-distributions η of thread configurations (C, σ). Thread-local transitions include transitions made by the thread itself and transitions made by the environment.

Let θ be a set of state transitions ($\theta \in \mathcal{P}(\mathit{State} \times \mathit{State})$). The transition made by the thread itself is represented as $\eta \hookrightarrow (\theta, \eta'')$, which is done in two steps. First we make the transition $\eta \rightsquigarrow (\theta, \eta')$ based on the concrete semantics and collect all state transitions in this step as θ, without considering splits. Then the splits in $\mathit{nextsplit}(\eta)$ are executed just like in the abstract operational semantics. Here $\mathit{nextsplit}(\eta)$ represents the set consisting of the next **split** statements to be executed in the thread configurations in $\mathit{supp}(\eta)$.

The transition made by the environment is represented as $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta''$, which is also done in two steps. The first step is represented as $\eta \overset{R}{\mapsto} \eta'$. We collect all transitions between thread configurations made by the environment as ψ. Note that the environment never modifies the thread's code and is assumed to make state transitions permitted in the rely condition R. The second step models the execution of split statements from the environment. No matter what split statement is executed by the environment, there always exists a b such that $\eta'|_b = \eta''$. If the environment does not execute any split statement, then η'' must be equal to η' and we can safely let b be true. In addition, the environment is assumed to perserve the invariant, i.e., $\eta'^{(\mathit{State})} \models I$.

$$
\begin{array}{rl}
\eta & \in \mathbb{D}_{\mathit{Stmt} \times \mathit{State}} \qquad \eta|_b \overset{\text{def}}{=} \eta|_{\lambda(C,\sigma).\sigma \models b} \\
\mathit{init}(C, \mu) & \overset{\text{def}}{=} \delta(C) \otimes \mu \qquad \mathit{nextsplit}(\eta) \overset{\text{def}}{=} \{\mathit{nextsplit}(C) \mid (C, \sigma) \in \mathit{supp}(\eta)\} \\
\psi & \in \mathcal{P}((\mathit{Stmt} \times \mathit{State}) \times (\mathit{Stmt} \times \mathit{State})) \\
\eta \overset{R}{\mapsto} \eta' & \text{iff } \exists \psi.\ \mathit{dom}(\psi) = \mathit{supp}(\eta) \wedge \mathit{range}(\psi) = \mathit{supp}(\eta') \wedge \\
& \qquad (\forall((C, \sigma), (C', \sigma')) \in \psi.\ C' = C \wedge (\sigma, \sigma') \models R) \\
\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'' & \text{iff } \exists \eta', b.\ \eta \overset{R}{\mapsto} \eta' \wedge \eta'|_b = \eta'' \wedge \eta''^{(\mathit{State})} \models I \\
\eta \rightsquigarrow (\theta, \eta') & \text{iff } \eta' = \lambda(C', \sigma').\textstyle\sum_{C,\sigma}\{p \cdot \eta(C, \sigma) \mid (C, \sigma) \xrightarrow{p} (C', \sigma')\} \wedge \\
& \qquad \theta = \{(\sigma, \sigma') \mid \exists C, C'.\ \eta(C, \sigma) > 0 \wedge (C, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\}
\end{array}
$$

$$
\frac{\eta \rightsquigarrow (\theta, \eta') \quad \mathit{nextsplit}(\eta) = \{\mathbf{split}(b_1, \ldots, b_k)\} \quad \eta'|_{b_i} = \eta''}{\eta \hookrightarrow (\theta, \eta'')}
$$

$$
\frac{\eta \rightsquigarrow \eta' \quad \#\mathit{nextsplit}(\eta) > 1}{\eta \hookrightarrow (\theta, \eta')}
$$

Fig. 28: Thread-local Transitions

The following definition describes that a thread behaves appropriately within n steps under the interference from the environment.

Definition 33. $(\eta, R, I) \Longrightarrow^{n}_{\text{ST}} (G, Q)$ is inductively defined as follows: $(\eta, R, I) \Longrightarrow^{0}_{\text{ST}} (G, Q)$ always holds; $(\eta, R, I) \Longrightarrow^{n+1}_{\text{ST}} (G, Q)$ if and only if the following are true:

1. $\eta^{(code)}(\mathbf{skip}) = 0$ or $\eta^{(code)}(\mathbf{skip}) = 1$.
2. if $\eta^{(code)}(\mathbf{skip}) > 0$, then $\eta^{(State)} \models Q$;
3. $\eta^{(State)} \models I$;
4. for all η', if $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longrightarrow^{n}_{\text{ST}} (G, Q)$;
5. for all θ, η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, $\eta'^{(State)} \models I$ and $(\eta', R, I) \Longrightarrow^{n}_{\text{ST}} (G, Q)$.

$(\eta, R, I) \Longrightarrow^{n}_{\text{ST}} (G, Q)$ means, from an initial distribution η, if the thread interfere with the environment within n steps where each environment step satifies R and preserves I, then each step done by the thread itself satifies G and preserves I. What's more, throughout this process, either all thread configurations in the support of the distribution is terminated or none of them is terminated, and if all thread configurations are terminated, then the state distribution satifies Q.

The semantics of the thread-local judgement $R, G, I \vdash_{\text{ST}} \{P\}C\{Q\}$ is defined below. Given the initial state distribution μ, the initial distribution of thread configurations is defined as $\mathit{init}(C, \mu)$ (see Fig. 28).

Definition 34. $R, G, I \models_{\text{ST}} \{P\}C\{Q\}$ iff for all μ, if $\mu \models I \wedge P$, then $(\mathit{init}(C, \mu), R, I) \Longrightarrow^{n}_{\text{ST}} (G, Q)$ for all n.

$R, G, I \models_{\text{ST}} \{P\}C\{Q\}$ means, from an inital state distribution satisfying I and P, the execution of C under the interference from the environment behaves appropriately within any finite number of steps.

Definition 35. $(\eta, R, I) \Longrightarrow^{n}_{\text{NST}} (G, Q)$ is inductively defined as follows: $(\eta, R, I) \Longrightarrow^{0}_{\text{NST}} (G, Q)$ always holds; $(\eta, R, I) \Longrightarrow^{n+1}_{\text{NST}} (G, Q)$ if and only if the following are true:

1. if $\eta^{(code)}(\mathbf{skip}) > 0$, then $\eta|_{\mathbf{skip}}{}^{(State)} \models Q$;
2. $\eta^{(State)} \models I$;
3. for all η', if $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longrightarrow^{n}_{\text{NST}} (G, Q)$;
4. for all θ, η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, $\eta'^{(State)} \models I$ and $(\eta', R, I) \Longrightarrow^{n}_{\text{NST}} (G, Q)$.

Here $\eta|_{\mathbf{skip}} \overset{\text{def}}{=} \eta|_{\lambda(C,\sigma).C=\mathbf{skip}}$ is the terminated part of η.

The main difference between $(\eta, R, I) \Longrightarrow^{n}_{\text{NST}} (G, Q)$ and $(\eta, R, I) \Longrightarrow^{n}_{\text{ST}} (G, Q)$ is that, for NST judgement, condition (1) it does not require η to be terminated or not terminated with probability 1. This way it allows "non-simultaneous

termination". To ensure that Q is still meaningful, it requires the terminated portion to satisfy Q.

The semantics of the thread-local judgement $R, G, I \vdash_{\text{NST}} \{P\}C\{Q\}$ is defined below.

Definition 36. $R, G, I \models_{\text{NST}} \{P\}C\{Q\}$ iff for all μ, if $\mu \models I \wedge P$, then $(\mathit{init}(C, \mu), R, I) \Longrightarrow^{n}_{\text{NST}} (G, Q)$ for all n.

Before giving the semantics of the $\vdash_{\text{SQ}}$ judgement, we define $[\![C]\!]$ as a function that maps an *initial state* σ to a sub-distribution of *final states*. We also lift the function to the distribution μ of the initial states. The definition of $[\![C]\!](\sigma)$ and $[\![C]\!](\mu)$ are similar to $[\![\mathbb{C}]\!]_{\varphi}(\sigma)$ and $[\![\mathbb{C}]\!]_{\varphi}(\mu)$ except that we don't need the scheduler for C is a sequential program.

$$[\![C]\!](\sigma) \stackrel{\text{def}}{=} \lambda\sigma'.\ \lim \vec{p}_{\sigma'}, \text{where } \forall n.\ (C, \sigma) \xrightarrow{\vec{p}_{\sigma'}[n]}{}^{n} (\mathbf{skip}, \sigma')$$
$$[\![C]\!](\mu) \stackrel{\text{def}}{=} \mathbb{E}_{\sigma\sim\mu}\{[\![C]\!](\sigma)\}$$

Then we can give the semantics of the sequential judgement $G \vdash_{\text{SQ}} \{P\}C\{Q\}$ below.

Definition 37. $G \models_{\text{SQ}} \{P\}C\{Q\}$ iff for all μ, if $\mu \models P$ and $|[\![C]\!](\mu)| = 1$, then $[\![C]\!](\mu) \models Q$ and for all $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$, $(\sigma, \sigma') \models G$.

Similar to Definition 9, the premise $|[\![C]\!](\mu)| = 1$ indicates the execution of C (with the initial state distribution μ) *terminates with the probability* 1. $G \models_{\text{SQ}} \{P\}C\{Q\}$ means, if the execution of C from an initial state distribution satifying P terminate with the probability 1, then the terminating state distribution satifies Q, and the transitions from any possible initial state to any possible terminating state reachable from that initial state satifies G.

G Proofs of More Examples

G.1 Shared 3-Sided Dice

As mentioned in Sec. 3.5, this algorithm does *not* work in SA, where different threads can be scheduled for different outcomes of coin flips. Consider such a strong adversary: It first nondeterministically selects a thread and keeps scheduling that thread until the thread gets heads on the coin flip, then it selects another thread and does the same thing, until all threads get heads. After that it schedules each thread with two consecutive steps such that each thread returns its own thread number.

We show the full proof of the shared 3-sided dice in Fig. 29, which we informally discussed in Sec. 3.5. We **split** atomically after *Roll* as before, but slightly revise the previously defined invariant I_{Dice}. This is because we need the invariant to be **lclosed** to apply the (PAR) rule, but $I_{Dice} = (\lceil x = 0 \rceil \vee (\lceil x \neq 0 \rceil \wedge \mathbb{E}(x) = 1))$

is not. A counterexample is the sequence $\overrightarrow{\mu}[n] = \delta(\{x \rightsquigarrow \frac{n+1}{2}\}) \oplus_{\frac{1}{n}} \delta(\{x \rightsquigarrow \frac{1}{2}\})$. Each $\overrightarrow{\mu}[n]$ in the sequence satisfies I_{Dice}, but its limit $\mu = \delta(\{x \rightsquigarrow \frac{1}{2}\})$ does not.

The main reason for I_{Dice} to be not **lclosed** is that x can be arbitrarily large. However, it is the formulation of I_{Dice} to blame, instead of the shared 3-sided dice which does enforce an upper bound for x. To see this, observe that each thread may increase x by setting it to 1, or doubling it when it is positive. The former trivially ensures x is bounded. The latter can be done at most once by each thread, thus, x should be bounded by 2^n, where n is the number of threads.

$$\frac{\dfrac{R_i, G_i, I \vdash \{\lceil x = 0\rceil\} C_i' \{\lceil x > 0\rceil\}}{\vdash_{\mathrm{A}} \{P\} C_1' \parallel \cdots \parallel C_n' \{Q\}}\ (\textsc{par})}{\vdash_{\mathrm{A}} \{P\} C_1 \parallel \cdots \parallel C_n \{Q\}}\ (\textsc{removesplit})$$

$$P \stackrel{\text{def}}{=} \lceil x = 0 \wedge (\forall i.\ c_i = 0)\rceil$$
$$Q \stackrel{\text{def}}{=} \lceil x > 0\rceil \wedge \mathbb{E}(x) = 1$$

$$R_i, G_i, I \vdash \begin{array}{l} \{\lceil x = 0\rceil\} \\ \{\lceil x = 0\rceil \vee \lceil x > 0\rceil\} \\ \textbf{while}\ (x = 0)\ \textbf{do} \\ \quad \{\lceil x = 0\rceil\} \\ \quad \{\lceil x = 0\rceil \vee \lceil x > 0 \wedge c_i = 0\rceil\} \\ \quad \langle \mathit{Roll}_i \rangle\ \textbf{split}(x = 0, x \neq 0) \\ \quad \{\lceil x = 0\rceil \vee \lceil x > 0\rceil\} \\ \{\lceil x > 0\rceil\} \end{array}$$

$$C_i \stackrel{\text{def}}{=} \textbf{while}\ (x = 0)\ \textbf{do}\ \langle \mathit{Roll}_i \rangle,$$
$$C_i' \stackrel{\text{def}}{=} \textbf{while}\ (x = 0)\ \textbf{do}\ \langle \mathit{Roll}_i \rangle\ \textbf{split}(x = 0, x \neq 0)$$
$$\mathit{Roll}_i \stackrel{\text{def}}{=} \langle x := 1 \rangle \oplus_{\frac{1}{2}} (\langle\ \mathit{SetFlag}(x, c_i)\ ; x := 2x \rangle \oplus_{\frac{1}{3}} \langle x := \tfrac{x}{2} \rangle)$$
$$\mathit{SetFlag}(x, c_i) \stackrel{\text{def}}{=} \textbf{if}\ (x > 0)\ \textbf{then}\ c_i := 1$$

$$I_{Dice} \stackrel{\text{def}}{=} \lceil x = 0\rceil \vee (\lceil x > 0\rceil \wedge \mathbb{E}(x) = 1)$$
$$I \stackrel{\text{def}}{=} I_{Dice} \wedge \lceil x \leq 2^{\sum_i c_i}\rceil \wedge (\forall i.\, \mathsf{IsFlag}(x, c_i))$$
$$R_i \stackrel{\text{def}}{=} ((x = 0 \ltimes x \geq 0) \vee (x > 0 \ltimes x > 0)) \wedge \mathbf{Inv}(c_i)$$
$$G_i \stackrel{\text{def}}{=} ((x = 0 \ltimes x \geq 0) \vee (x > 0 \ltimes x > 0)) \wedge (\forall j \neq i. \mathbf{Inv}(c_j))$$
$$\mathsf{IsFlag}(x, c_i) \stackrel{\text{def}}{=} \lceil c_i = 0 \vee c_i = 1\rceil \wedge \lceil x = 0 \Rightarrow c_i = 0\rceil$$

Fig. 29: Proof Sketch of Shared 3-sided Dice.

To formulate this observation, we introduce an auxiliary flag c_i for each thread i, to record whether thread i has doubled a positive x. We instrument the loop body Roll_i with an auxiliary statement $\mathit{SetFlag}(x, c_i)$ just before $x := 2x$, that sets c_i when a positive x is doubled by thread i. Meanwhile, we strengthen I_{Dice} into I, as shown in the bottom half of Fig. 29. I additionally requires $\lceil x \leq 2^{\sum_i c_i}\rceil$ and $\mathsf{IsFlag}(x, c_i)$. The former enforces an upper bound on x, and the latter enforces the integrity of the values stored in c_i. Now we can prove **lclosed**(I), as explained in Sec. 5.1.

The rest of the proof is straightforward using our logic rules, sketched in Fig. 29. The precondition P additionally require c_i to be initialized to 0. The rely and guarantee conditions R_i and G_i simply says x changes only from 0 to

positive, and stays positive thereafter. They additionally require that c_i is local to each thread using $\mathbf{Inv}(c_i)$ (see Fig. 13 for the definition of $\mathbf{Inv}$).

The loop body $\langle \mathit{Roll}_i \rangle$ $\mathbf{split}(x = 0, x \neq 0)$ is verified by applying the (ATOM-SPLIT) rule, where Q is instantiated with $(\lceil x = 0 \rceil \vee \lceil x > 0 \rceil)$. This requires us to verify Roll_i:

$$G_i \vdash_{\text{SQ}} \{I \wedge (\lceil x = 0 \rceil \vee \lceil x > 0 \wedge c_i = 0 \rceil)\} \mathit{Roll}_i \{(I \wedge Q \wedge \lceil x = 0 \rceil) \oplus (I \wedge Q \wedge \lceil x \neq 0 \rceil)\} \ .$$

It is equivalent to the following, which simplifies the postcondition and can be proved by applying the (SQ-PCH) rule:

$$G_i \vdash_{\text{SQ}} \{I \wedge (\lceil x = 0 \rceil \vee \lceil x > 0 \wedge c_i = 0 \rceil)\} \mathit{Roll}_i \{(I \wedge \lceil x = 0 \rceil) \oplus (I \wedge \lceil x \neq 0 \rceil)\} \ .$$

G.2 Conciliator

As introduced in Sec. 1, [12] gives a probabilistic-write based conciliator for *probabilistic agreement* between n threads, each thread i executing C_i below.

$$C_i \stackrel{\text{def}}{=} (\mathbf{while}\ (s = 0)\ \mathbf{do}\ \langle s := i \rangle \oplus_p \langle \mathbf{skip} \rangle)\,; y_i := s$$

Here s is a shared variable, y_i is the local variable for thread i that records its return value.

We want to prove $\{\lceil s = 0 \rceil\} C_1 \parallel \cdots \parallel C_n \{\mathbf{Pr}(y_1 = \cdots = y_n) \geq (1-p)^{n-1}\}$. Intuitively the postcondition holds because, when there is exactly one thread i succeeded in writing to s, all threads will return i. This ideal case happens with probability no less than $(1-p)^{n-1}$ in OA, because (i) for the program to terminate, at least one thread has updated s, and (ii) after the first update to s, each of the other $n-1$ threads has at most one chance to update s, and such an update happens with probability no more than $1-p$. Note that this algorithm does *not* work in SA, where different threads can be scheduled for different outcomes of coin flips. Consider such a strong adversary: It first nondeterministically selects a thread and keeps scheduling that thread until the thread gets heads on the coin flip, then it selects another thread and does the same thing, until all threads get heads. After that it schedules each thread with two consecutive steps such that each thread returns its own thread number.

To formulate the intuition, we introduce a shared auxiliary variable c that counts how many threads have written to s and insert the auxiliary code $c := c{+}1$ which is executed atomically with $s := i$ (see PWrite_i in Fig. 30). Similar to the shared 3-sided dice example, we also introduce flag variables d_i to formalize the "at most one chance" update to s. When d_i is set, it means thread i can no longer update s. We insert the auxiliary code $\mathit{SetFlag}(s, d_i)$ to set d_i at the proper time.

The proof is sketched in Fig. 30. At the whole-program level, we apply the (REMOVESPLIT) and (LAZYCOIN) rules to wrap the probabilistic choice in an atomic block, and to instrument $\mathbf{split}(s = 0, s \neq 0)$ after the loop body such that the resulting smaller distributions either enter or exit the loop. Using the (PAR) rule, our goal becomes to thread-locally verify C_i'' in Fig. 30. In thread-local proof, the invariant I says that either $s = 0$ (and thus $c = 0$ and each thread has chance to update s), or $s \neq 0$ (and thus $c > 0$) and the probability

$$\frac{\dfrac{\dfrac{R_i, G_i, I \vdash \{P_i\} C_i'' \{Q_i\}}{\vdash_{\mathrm{A}} \{P\} C_1'' \parallel \cdots \parallel C_n'' \{Q\}}\ (\textsc{par})}{\vdash_{\mathrm{A}} \{P\} C_1' \parallel \cdots \parallel C_n' \{Q\}}\ (\textsc{removesplit})}{\vdash_{\mathrm{A}} \{P\} C_1 \parallel \cdots \parallel C_n \{Q\}}\ (\textsc{lazycoin})$$

$P \stackrel{\text{def}}{=} \lceil c = 0 \wedge s = 0 \wedge (\forall i. d_i = 0) \rceil$
$Q \stackrel{\text{def}}{=} \mathbf{Pr}(y_1 = \cdots = y_n) \geq (1-p)^{n-1}$

$R_i, G_i, I \vdash$
$\{\lceil s = 0 \rceil \vee \lceil s \neq 0 \rceil\}$
while $(s = 0)$ **do**
  $\{\lceil s = 0 \rceil\}$
  $\{\lceil s = 0 \rceil \vee \lceil s \neq 0 \wedge d_i = 0 \rceil\}$
  skip;
  $\{\lceil s = 0 \rceil \vee \lceil s \neq 0 \wedge d_i = 0 \rceil\}$
  $\langle PWrite_i \rangle$ **split**$(s = 0, s \neq 0)$;
  $\{\lceil s = 0 \rceil \vee \lceil s \neq 0 \rceil\}$
$\{\lceil s \neq 0 \rceil\}$
$\langle\ SetFlag(s, d_i)\ ;\ y_i := s \rangle$
$\{\lceil d_i = 1 \wedge ((c = 1 \wedge y_i = s) \vee (c > 1)) \rceil\}$

$C_i \stackrel{\text{def}}{=} (\textbf{while}\ (s = 0)\ \textbf{do}\ PWrite_i); \langle\ SetFlag(s, d_i)\ ; y_i := s \rangle$
$C_i' \stackrel{\text{def}}{=} (\textbf{while}\ (s = 0)\ \textbf{do}\ (\textbf{skip}; \langle PWrite_i \rangle)); \langle\ SetFlag(s, d_i)\ ; y_i := s \rangle$
$C_i'' \stackrel{\text{def}}{=} (\textbf{while}\ (s = 0)\ \textbf{do}\ (\textbf{skip}; \langle PWrite_i \rangle\ \textbf{split}(s = 0, s \neq 0))); \langle\ SetFlag(s, d_i)\ ; y_i := s \rangle$
$PWrite_i \stackrel{\text{def}}{=} \langle s := i;\ c := c + 1; SetFlag(s, d_i)\ \rangle \oplus_p \langle \textbf{skip};\ SetFlag(s, d_i)\ \rangle$

$I \stackrel{\text{def}}{=} \lceil s = 0 \wedge c = 0 \wedge (\forall i.\, d_i = 0) \rceil \vee (\lceil s \neq 0 \wedge c > 0 \rceil \wedge \mathsf{PBound})$
$\mathsf{PBound} \stackrel{\text{def}}{=} \exists K \leq n. \lceil (\forall i.\, d_i = 0 \vee d_i = 1) \wedge \sum_{i=1}^{n} d_i = K \rceil \wedge \mathbf{Pr}(c = 1) \geq (1-p)^{K-1}$
$P_i \stackrel{\text{def}}{=} \lceil s = 0 \rceil \vee \lceil s \neq 0 \rceil, \qquad Q_i \stackrel{\text{def}}{=} \lceil d_i = 1 \wedge ((c = 1 \wedge y_i = s) \vee (c > 1)) \rceil$
$R_i \stackrel{\text{def}}{=} ((\mathbf{Inv}(c) \wedge \mathbf{Inv}(s)) \vee (\exists N.\, c = N \ltimes (c = N + 1 \wedge s \neq 0))) \wedge \mathbf{Inv}(d_i) \wedge \mathbf{Inv}(y_i)$
$G_i \stackrel{\text{def}}{=} ((\mathbf{Inv}(c) \wedge \mathbf{Inv}(s)) \vee (\exists N.\, c = N \ltimes (c = N + 1 \wedge s \neq 0))) \wedge (\forall j \neq i. \mathbf{Inv}(d_j) \wedge \mathbf{Inv}(y_j))$

Fig. 30: Proof Sketch of Conciliator.

of $c = 1$ has a lower bound (specified by PBound). The rest of the proof follows directly from our logic rules.

G.3 Multiplayer Level-up Game

We verify the multiplayer level-up game using the instrumented code $\widehat{\mathbb{C}_{LvUp}} = \widehat{LvUp_1} \parallel \cdots \parallel \widehat{LvUp_n}$, where the code for thread i is $\widehat{LvUp_i}$:

1 $k_i = 1$;
2 **while** $(k_i \leq m \wedge v_i = 0)$ **do**
3 $\langle x[k_i] := x[k_i] + 1;\ y[k_i] := y[k_i] + 1; z_i[k_i] := 1; w_i[k_i] := 1\ \rangle$
  $\oplus_p\ \langle v_i := 1;\ y[k_i] := y[k_i] + 1; w_i[k_i] := 1\ \rangle$
4 $k_i := k_i + 1$;

Following the intuition in Appendix E, we introduce auxiliary array $y[1..m]$ such that $y[j]$ records the number of threads that executes line 3 of round j, and instrument line 3 of $LvUp_i$ with auxiliary code $y[k_i] := y[k_i] + 1$. We further introduce auxiliary thread local arrays $w_i[1..m]$ and $z_i[1..m]$ for the convenience

of local reasoning. Here $w_i[j]$ is a flag recording whether thread i has executed line 3 of round j, and $z_i[j]$ is a flag recording whether thread i has leveled up in round j (execute the left branch with probability p). Auxiliary code for setting w_i and z_i are also appended in line 3.

The proof sketch is at the top of Fig. 31. As in Sec. E, we prove post conditions Q_j for each round j by applying (P-CSQ) and (BIGCONJ) rules. We use the NST judgements for thread local proofs, and insert $\mathbf{split}(k_i < j \wedge v_i = 0, k_i < j \wedge v_i = 1, k_i \geq j)$ at line 3 of $\widehat{LvUp_i}$. We denote the resulting thread local code by $C_i'(j)$. In addition to $\mathbb{E}(x[j]) = p \cdot \mathbb{E}(y[j])$, the invariant I_j further require that the flags $z_i[j]$ and $w_i[j]$ are consistent with $x[j]$ and $y[j]$, i.e., $\lceil x[j] = \sum_i z_i[j] \wedge y[j] = \sum_i w_i[j] \rceil$.

The rest of the proof is straightforward. The seemingly complicated intermediate assertions such as T_{ji} and M_{ji} are mostly talking about the data consistencies of the auxiliary variables.

G.4 Group Election

Group election [2] is a probabilistic algorithm for consensus on leadership. It selects a relatively small group of leaders rapidly against oblivious adversaries.

A simplified version of group election $\mathbb{C}_{Elct}$ consists of n threads, each thread runs the following code $Elct_i$:

```
1  k_i = 1;
2  while (k_i ≤ m ∧ v_i = 0) do
3     ⟨s[k_i] := 1⟩ ⊕_p ⟨v_i := s[k_i]⟩;
4     k_i := k_i + 1;
5  y_i := 1 − v_i;
```

Similar to $\mathbb{C}_{LvUp}$, the election has m rounds. In the shared array $s[1..m]$, each $s[j]$ records whether there is a winner in round j. In each round j, the threads try to win the round and progress to the next round, by writing 1 to $s[j]$ with probability p. Different from $\mathbb{C}_{LvUp}$, if thread i failed in round j, it does not exit the election immediately, but checks whether someone else has won by reading $s[j]$ into v_i. If so ($v_i = 1$), thread i terminates and lose the election; if not ($v_i = 0$), thread i automatically progress to the next round. In the end, a thread with $v_i = 0$ becomes a leader and sets its local variable y_i to 1, while a thread with $v_i = 1$ becomes a follower and set its local variable y_i to 0.

We want to verify that $\mathbb{C}_{Elct}$ satisfies the postcondition $\mathbb{E}(\sum_i y_i) \leq f^m(n)$, where $f = \lambda x.p \cdot x + \frac{1}{p}$. To see why this holds, we first observe an invariant that, for any round j,

$$\mathbb{E}(\#\text{threads finish line 3 with } v_i = 0) \leq f\,(\mathbb{E}(\#\text{threads execute line 3})).$$

This follows by observing that (i) in line 3, a thread has probability p to win, and (ii) calculating the number of thread that execute the right branch in line 3 and return $v_i = 0$, is the same as calculating the time before success of a finite number of independent Bernoulli trial with success probability p, which has an expectation no larger than $\frac{1}{p}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{R_i, G_i, I_j \vdash_{\text{NST}} \{P_i\}C_i'(j)\{Q_{ji}\}}{\vdash_{\text{A}} \{P\}C_1'(j) \parallel \cdots \parallel C_n'(j)\{Q_j\}}\ (\text{PAR})}{\vdash_{\text{A}} \{P\}\widehat{\mathbb{C}_{LvUp}}\{Q_j\}}\ (\text{REMOVESPLIT})}{\vdash_{\text{A}} \{P \wedge \cdots \wedge P\}\widehat{\mathbb{C}_{LvUp}}\{Q_1 \wedge \cdots \wedge Q_m\}}\ (\text{BIGCONJ})}{\vdash_{\text{A}} \{P\}\widehat{\mathbb{C}_{LvUp}}\{Q\}}\ (\text{P-CSQ})$$

$R_i, G_i, I_j \vdash_{\text{NST}}$

$\{(\exists K \le j.\ \lceil k_i = K \wedge v_i = 0 \wedge T_{ji}\rceil) \vee \lceil (k_i > j \vee v_i = 1) \wedge T_{ji}\rceil\}$
while $(k_i \le m \wedge v_i = 0)$ **do**
  $\{\exists K \le j.\ \lceil k_i = K \wedge v_i = 0 \wedge T_{ji}\rceil\}$
  $\{(\lceil k_i = j \wedge v_i = 0 \wedge T_{ji}\rceil) \vee (\exists K < j.\ \lceil k_i = K \wedge v_i = 0 \wedge T_{ji}\rceil)\}$
  $\langle (x[k_i] := x[k_i] + 1;\ y[k_i] := y[k_i] + 1; z_i[k_i] := 1; w_i[k_i] := 1\ )\rangle$
  $\oplus_p$
  $\langle v_i := 1;\ y[k_i] := y[k_i] + 1; w_i[k_i] := 1\ \rangle$
  split$(k_i < j \wedge v_i = 0, k_i < j \wedge v_i = 1, k_i \ge j)$
  $\{\lceil k_i = j \wedge M_{ji}\rceil \vee (\exists K < j.\ \lceil k_i = K \wedge v_i = 0 \wedge M_{ji}\rceil) \vee$
    $(\exists K < j.\ \lceil (k_i = K \wedge v_i = 1) \wedge M_{ji}\rceil)\}$
  $k_i := k_i + 1$
  $\{(\exists K \le j.\ \lceil k_i = K \wedge v_i = 0 \wedge T_{ji}\rceil) \vee \lceil (k_i > j \vee v_i = 1) \wedge T_{ji}\rceil\}$
$\{\lceil T_{ji} \wedge \lceil k_i > m \vee v_i \ne 0\rceil\rceil\}$
$\{\lceil w_i[1] = 1 \wedge (j < m \Rightarrow z_i[j] = w_i[j+1])\rceil\}$

$$\begin{aligned}
P &\stackrel{\text{def}}{=} \lceil (\forall i.\ k_i = 1 \wedge v_i = 0) \wedge (\forall j.\ x[j] = 0 \wedge y[j] = 0)\rceil \\
Q &\stackrel{\text{def}}{=} \forall j \le m.\ \mathbb{E}(x[j]) = n \cdot p^j \\
Q_j &\stackrel{\text{def}}{=} \mathbb{E}(x[j]) = p \cdot \mathbb{E}(y[j]) \wedge \lceil y[1] = n \wedge (j < m \Rightarrow x[j] = y[j+1])\rceil \\
I_j &\stackrel{\text{def}}{=} \mathbb{E}(x[j]) = p \cdot \mathbb{E}(y[j]) \wedge \lceil x[j] = \textstyle\sum_i z_i[j] \wedge y[j] = \sum_i w_i[j]\rceil \\
P_i &\stackrel{\text{def}}{=} (\exists K \le j.\ \lceil k_i = K \wedge v_i = 0 \wedge T_{ji}\rceil) \vee \lceil (k_i > j \vee v_i = 1) \wedge T_{ji}\rceil \\
Q_{ji} &\stackrel{\text{def}}{=} \lceil w_i[1] = 1 \wedge (j < m \Rightarrow z_i[j] = w_i[j+1])\rceil \\
L_i &\stackrel{\text{def}}{=} (k_i = 1 \wedge v_i = 0) \vee w_i[1] = 1 \\
T_{ji} &\stackrel{\text{def}}{=} L(i) \wedge \left(\begin{array}{l}(k_i \le j \wedge z_i[j] = 0 \wedge w_i[j+1] = 0) \vee \\ (k_i = j+1 \wedge w_i[j+1] = 0 \wedge \\ \quad ((z_i[j] = 1 \wedge v_i = 0) \vee (z_i[j] = 0 \wedge v_i = 1))) \vee \\ (k_i > j+1 \wedge z_i[j] = 1 \wedge w_i[j+1] = 1))\end{array}\right) \\
M_{ji} &\stackrel{\text{def}}{=} L(i) \wedge \left(\begin{array}{l}(k_i < j \wedge z_i[j] = 0 \wedge w_i[j+1] = 0) \vee \\ (k_i = j \wedge w_i[j+1] = 0 \wedge \\ \quad ((z_i[j] = 1 \wedge v_i = 0) \vee (z_i[j] = 0 \wedge v_i = 1))) \vee \\ (k_i \ge j+1 \wedge z_i[j] = 1 \wedge w_i[j+1] = 1))\end{array}\right) \\
R_i &\stackrel{\text{def}}{=} \mathbf{Inv}(k_i) \wedge \mathbf{Inv}(v_i) \wedge (\forall j.\ \mathbf{Inv}(z_i[j]) \wedge \mathbf{Inv}(w_i[j])) \\
G_i &\stackrel{\text{def}}{=} \textstyle\bigwedge_{i' \ne i} R_{i'}
\end{aligned}$$

Fig. 31: Proof Sketch of Multiplayer Level-up Game

The proof is sketched in Fig. 32. We introduce two thread local auxiliary arrays $c_i[1..m]$ and $w_i[1..m]$ to formalize the invariant. For each thread i and round j, $c_i[j]$ records whether thread i has executed line 3 in round j, $w_i[j]$ records whether thread i finish round j with $v_i = 0$. The previously introduced intuition is formalized in a more precise way:

$$\mathbb{E}(\sum_{i=1}^{n} w_i[j]) = \mathbb{E}(p \cdot \sum_{i=1}^{n} c_i[j] + \frac{1-p-(1-p)^{1+\sum_{i=1}^{n} c_i[j]}}{p}).$$

The above assertion is the main part of S_j, which is the main part the invariant $I_j = S_j \mid s[1], \ldots, s[j-1]$. Here $P \mid x_1, \ldots, x_n \stackrel{\text{def}}{=} \bigoplus(\exists X_1, \ldots, X_n. \lceil \forall i. x_i = X_i \rceil \wedge P)$, it implies P is independent of $x_1, \ldots, x_n$.

$$\dfrac{\dfrac{\dfrac{\forall i \le n.\ R_i, G_i, I_j \vdash_{\text{NST}} \{P\}C_i'(j)\{Q_{ji}\}}{\vdash_{\text{A}} \{P\}C_1'(j) \parallel \cdots \parallel C_n'(j)\{Q_j\}}\ (\text{PAR})}{\vdash \{P\}C_1 \parallel \cdots \parallel C_n\{Q_j\}}\ (\text{REMOVESPLIT})}{\vdash \{P \wedge \cdots \wedge P\}C_1 \parallel \cdots \parallel C_n\{Q_1 \wedge \cdots \wedge Q_m\}}\ (\text{BIGCONJ})}{\vdash \{P\}C_1 \parallel \cdots \parallel C_n\{\mathbb{E}(\sum_{i=1}^n y_i) \le f^m(n)\}}\ (\text{P-CSQ})$$

$R_i, G_i, I_j \vdash_{\text{NST}}$

$\{P\}$
$\{(\exists K \le j.\ \lceil k_i = K \wedge v_i = 0 \wedge T_{ji}\rceil) \vee \lceil (k_i > j \vee v_i = 1) \wedge T_{ji}\rceil\}$
while $(k_i \le m \wedge v_i = 0)$ **do**
  $\{\exists K \le j.\ \lceil k_i = K \wedge v_i = 0 \wedge T_{ji}\rceil\}$
  $\{(\lceil k_i = j \wedge v_i = 0 \wedge T_{ji}\rceil) \vee (\exists K < j.\ \lceil k_i = K \wedge v_i = 0 \wedge T_{ji}\rceil)\}$
  $\langle (s[k_i] := 1;\ c_i[k_i] := 1; w_i[k_i] := 1\ ) \oplus_p$
  $(v_i := s[k_i];\ c_i[k_i] := 1;\ \textbf{if}\ (v_i = 0)\ \textbf{then}\ w_i[k_i] := 1\ )\rangle$
  split$(k_i < j \wedge v_i = 0, k_i < j \wedge v_i \neq 0, k_i \ge j)$;
  $\{\lceil k_i = j \wedge M_{ji}\rceil \vee (\exists K < j.\ \lceil k_i = K \wedge v_i = 0 \wedge M_{ji}\rceil) \vee$
  $(\exists K < j.\ \lceil (k_i = K \wedge v_i = 1) \wedge M_{ji}\rceil)\}$
  $k_i := k_i + 1$
  $\{(\exists K \le j.\ \lceil k_i = K \wedge v_i = 0 \wedge T_{ji}\rceil) \vee \lceil (k_i > j \vee v_i = 1) \wedge T_{ji}\rceil\}$
$\{\lceil T_{ji} \wedge \lceil k_i > m \vee v_i \neq 0\rceil\rceil\}$
$\{\lceil ((w_i[m] = 1 \wedge v_i = 0) \vee (w_i[m] = 0 \wedge v_i = 1)) \wedge$
  $c_i[1] = 1 \wedge (j < m \Rightarrow w_i[j] = c_i[j+1])\rceil\}$
$\langle y_i := 1 - v_i \rangle$
$\{\lceil ((w_i[m] = 1 \wedge y_i = 1) \vee (w_i[m] = 0 \wedge y_i = 0)) \wedge$
  $c_i[1] = 1 \wedge (j < m \Rightarrow w_i[j] = c_i[j+1])\rceil\}$
$\{\lceil y_i = w_i[m] \wedge c_i[1] = 1 \wedge (j < m \Rightarrow w_i[j] = c_i[j+1])\rceil\}$

$$\begin{aligned}
Q_j &\stackrel{\text{def}}{=} \mathbb{E}(\textstyle\sum_{i=1}^n w_i[j]) \le p \cdot \mathbb{E}(\sum_{i=1}^n c_i[j]) + \frac{1}{p} \wedge \\
&\quad \lceil \forall i.\ y_i = w_i[m] \wedge c_i[1] = 1 \wedge (j < m \Rightarrow w_i[j] = c_i[j+1])\rceil \\
Q_{ji} &\stackrel{\text{def}}{=} \lceil y_i = w_i[m] \wedge c_i[1] = 1 \wedge (j < m \Rightarrow w_i[j] = c_i[j+1])\rceil \\
S_j &\stackrel{\text{def}}{=} \mathbb{E}(\textstyle\sum_{i=1}^n w_i[j]) = \mathbb{E}(p \cdot \sum_{i=1}^n c_i[j] + \frac{1-p-(1-p)^{1+\sum_{i=1}^n c_i[j]}}{p}) \wedge \\
&\quad \mathbf{Pr}(s[j] = 0) = \mathbb{E}((1-p)^{\sum_{i=1}^n c_i[j]}) \\
I_j &\stackrel{\text{def}}{=} S_j \mid s[1], \ldots, s[j-1] \\
P &\stackrel{\text{def}}{=} \lceil (\forall i.\ k_i = 1 \wedge v_i = 0) \wedge (\forall j.\ s[j] = 0 \wedge y[j] = 0) \wedge (\forall i, j.\ w_i[j] = 0 \wedge c_i[j] = 0)\rceil \\
L_i &\stackrel{\text{def}}{=} ((w_i[m] = 1 \wedge v_i = 0) \vee (w_i[m] = 0 \wedge (k_i \le m \vee v_i = 1))) \wedge \\
&\quad ((k_i = 1 \wedge v_i = 0) \vee c_i[1] = 1) \\
T_{ji} &\stackrel{\text{def}}{=} L_i \wedge \begin{pmatrix} (k_i \le j \wedge w_i[j] = 0 \wedge c_i[j+1] = 0) \vee \\ (k_i = j+1 \wedge c_i[j+1] = 0 \wedge \\ \quad ((w_i[j] = 1 \wedge v_i = 0) \vee (w_i[j] = 0 \wedge v_i = 1))) \vee \\ (k_i > j+1 \wedge w_i[j] = 1 \wedge c_i[j+1] = 1) \end{pmatrix} \\
H_i &\stackrel{\text{def}}{=} ((w_i[m] = 1 \wedge v_i = 0) \vee (w_i[m] = 0 \wedge (k_i < m \vee (k_i = m \wedge v_i = 1)))) \wedge \\
&\quad ((k_i = 1 \wedge v_i = 0) \vee c_i[1] = 1) \\
M_{ji} &\stackrel{\text{def}}{=} H(i) \wedge \begin{pmatrix} (k_i < j \wedge w_i[j] = 0 \wedge c_i[j+1] = 0) \vee \\ (k_i = j \wedge c_i[j+1] = 0 \wedge \\ \quad ((w_i[j] = 1 \wedge v_i = 0) \vee (w_i[j] = 0 \wedge v_i = 1))) \vee \\ (k_i \ge j+1 \wedge w_i[j] = 1 \wedge c_i[j+1] = 1) \end{pmatrix} \\
R_i &\stackrel{\text{def}}{=} \mathbf{Inv}(k_i) \wedge \mathbf{Inv}(v_i) \wedge \mathbf{Inv}(y_i) \wedge (\forall j.\ \mathbf{Inv}(c_i[j]) \wedge \mathbf{Inv}(w_i[j])) \\
G_i &\stackrel{\text{def}}{=} \textstyle\bigwedge_{j \neq i} R_i
\end{aligned}$$

Fig. 32: Proof Sketch of Group Election

H Proof of Soundness

H.1 Preliminary Lemmas

Lemma 1. *For all set A and $\mu \in \mathbb{D}_A$, $E_1, \ldots, E_k \in A \to Prop$, if $\forall i, j.\ i \neq j \implies \neg(E_i(a) \wedge E_j(a))$ for all $a \in A$, then $\boldsymbol{Pr}_{a\in\mu}[E_1(a) \vee \cdots \vee E_k(a)] = \sum_{i=1}^{k} \boldsymbol{Pr}_{a\in\mu}[E_i(a)]$.*

Proof. For all set A and $\mu \in \mathbb{D}_A$, $E_1, \ldots, E_k \in A \to \mathrm{Prop}$ such that $\forall i, j.\ i \neq j \implies \neg(E_i(a) \wedge E_j(a))$ for all $a \in A$, we have $\mathbf{Pr}_{a\in\mu}[E_1(a) \vee \cdots \vee E_k(a)] = \sum_a \{\mu(a) \mid E_1(a) \vee \cdots \vee E_k(a)\} = \sum_{i=1}^{k} \mathbf{Pr}_{a\in\mu}[E_i(a)]$.

Lemma 2. *For all set A, B and $\mu \in \mathbb{D}_{A\times B}$, $E \in A \to Prop$, $\boldsymbol{Pr}_{(a,b)\sim\mu}[E(a)] = \boldsymbol{Pr}_{a\sim\mu^{(A)}}[E(a)]$.*

Proof. For all set A, B and $\mu \in \mathbb{D}_{A\times B}$, $E \in A \to \mathrm{Prop}$,

$$\begin{aligned} &\mathbf{Pr}_{(a,b)\sim\mu}[E(a)] \\ &= \textstyle\sum_{a,b}\{\mu(a,b) \mid E(a)\} \\ &= \textstyle\sum_a\{\sum_b \mu(a,b) \mid E(a)\} \\ &= \textstyle\sum_a\{\mu^{(A)}(a) \mid E(a)\} \\ &= \mathbf{Pr}_{a\sim\mu^{(A)}}[E(a)]. \end{aligned}$$

Lemma 3. *For all set A, B and $\mu \in \mathbb{D}_{A\times B}$, $E \in B \to Prop$, $\boldsymbol{Pr}_{(a,b)\sim\mu}[E(b)] = \boldsymbol{Pr}_{b\sim\mu^{(B)}}[E(b)]$.*

Proof. For all set A, B and $\mu \in \mathbb{D}_{A\times B}$, $E \in B \to \mathrm{Prop}$,

$$\begin{aligned} &\mathbf{Pr}_{(a,b)\sim\mu}[E(b)] \\ &= \textstyle\sum_{a,b}\{\mu(a,b) \mid E(b)\} \\ &= \textstyle\sum_b\{\sum_a \mu(a,b) \mid E(b)\} \\ &= \textstyle\sum_b\{\mu^{(B)}(b) \mid E(b)\} \\ &= \mathbf{Pr}_{b\sim\mu^{(B)}}[E(b)]. \end{aligned}$$

Lemma 4. *For all set A and $\mu \in \mathbb{D}_A, E \in A \to Prop$, if $\boldsymbol{Pr}_{a\sim\mu}[E(a)] = 1$, then $\mu|_E = \mu$.*

Proof. For all set A and $\mu \in \mathbb{D}_A, E \in A \to \mathrm{Prop}$ such that $\mathbf{Pr}_{a\sim\mu}[E(a)] = 1$, we have $\sum_a\{a \mid E(a)\} = \mathbf{Pr}_{a\sim\mu}[E(a)] = 1 = |\mu| = \sum_a \mu(a) = \sum_a\{a \mid E(a)\} + \sum_a\{a \mid \neg E(a)\}$, thus $\sum_a\{a \mid \neg E(a)\} = 0$, so $\mu(a) = 0$ for all a such that $E(a)$ does not hold. Therefore,

$$\mu|_E = \lambda a. \begin{cases} \frac{\mu(a)}{\mathbf{Pr}_{a\sim\mu}E(a)}, & \text{if } E(a) \\ 0, & \text{otherwise} \end{cases} = \lambda a. \begin{cases} \mu(a), & \text{if } E(a) \\ \mu(a), & \text{otherwise} \end{cases} = \mu.$$

Lemma 5. *For all set A and $\mu \in \mathbb{D}_A$, $\mu|_{\lambda a\in A.true} = \mu$.*

Proof. For all set A and $\mu \in \mathbb{D}_A$, from $\mathbf{Pr}_{a\sim\mu}[\mathrm{true}] = \sum_a \mu(a) = |\mu| = 1$ by Lem. 4 we know $\mu|_{\lambda a\in A.\mathrm{true}} = \mu$.

Lemma 6. *Let A be any set, $\vec{\mu}$ be any infinite sequence of sub-distributions over A, and μ be any sub-distribution over A, if $\lim \vec{\mu} = \mu$, then for all $a \in A$, $\lim_{n\to\infty} \vec{\mu}[n](a) = \mu(a)$.*

Proof. From $\lim \vec{\mu} = \mu$ we know $\lim_{n\to\infty} \sum_{a'\in A} |\vec{\mu}[n](a') - \mu(a')| = 0$. For all a, to prove $\lim_{n\to\infty} \vec{\mu}[n](a) = \mu(a)$, we need to prove for all $\epsilon > 0$, there exists N such that $|\vec{\mu}[n](a) - \mu(a)| < \epsilon$ for all $n \geq N$. For all $\epsilon > 0$, from $\lim_{n\to\infty} \sum_{a'\in A} |\vec{\mu}[n](a') - \mu(a')| = 0$ we know there exists N such that for all $n \geq N$, $\sum_{a'\in A} |\vec{\mu}[n](a') - \mu(a')| < \epsilon$, thus $|\vec{\mu}[n](a) - \mu(a)| < \epsilon$ for all $n \geq N$.

Definition 38. Let A be any set, $\vec{\mu}$ be any infinite sequence of sub-distributions over A, and $f \in A \to \mathbb{D}_B$, we define $f(\vec{\mu}) \stackrel{\text{def}}{=} \lambda n.\ f(\vec{\mu}[n])$.

Lemma 7. *Let A, B be any set, $\vec{\mu}$ be any infinite sequence of sub-distributions over $A \times B$ and μ be any sub-distribution over $A \times B$, if $\lim \vec{\mu} = \mu$, then $\lim \vec{\mu}^{(A)} = \mu^{(A)}$.*

Proof. From $\lim \vec{\mu} = \mu$ we know $\lim_{n\to\infty} \sum_{a\in A, b\in B} |\vec{\mu}[n](a,b) - \mu(a,b)| = 0$, thus

$$\begin{aligned}
&\lim_{n\to\infty} \textstyle\sum_{a\in A} |\vec{\mu}[n]^{(A)}(a) - \mu^{(A)}(a)| \\
&= \lim_{n\to\infty} \textstyle\sum_{a\in A} |\sum_{b\in B} \vec{\mu}[n](a,b) - \sum_{b\in B} \mu(a,b)| \\
&= \lim_{n\to\infty} \textstyle\sum_{a\in A} |\sum_{b\in B} (\vec{\mu}[n](a,b) - \mu(a,b))| \\
&\leq \lim_{n\to\infty} \textstyle\sum_{a\in A} \sum_{b\in B} |\vec{\mu}[n](a,b) - \mu(a,b)| \\
&= 0.
\end{aligned}$$

From $\sum_{a\in A} |\vec{\mu}[n]^{(A)}(a) - \mu^{(A)}(a)| \geq 0$ for all n we know $\lim_{n\to\infty} \sum_{a\in A} |\vec{\mu}[n]^{(A)}(a) - \mu^{(A)}(a)| \geq 0$, thus $\lim_{n\to\infty} \sum_{a\in A} |\vec{\mu}[n]^{(A)}(a) - \mu^{(A)}(a)| = 0$. Therefore $\lim \vec{\mu}^{(A)} = \mu^{(A)}$.

Lemma 8. *Let A, B be any set, $\vec{\mu}$ be any infinite sequence of sub-distributions over $A \times B$ and μ be any sub-distribution over $A \times B$, if $\lim \vec{\mu} = \mu$, then $\lim \vec{\mu}^{(B)} = \mu^{(B)}$.*

Proof. From $\lim \vec{\mu} = \mu$ we know $\lim_{n\to\infty} \sum_{a\in A, b\in B} |\vec{\mu}[n](a,b) - \mu(a,b)| = 0$, thus

$$\begin{aligned}
&\lim_{n\to\infty} \textstyle\sum_{b\in B} |\vec{\mu}[n]^{(B)}(b) - \mu^{(B)}(b)| \\
&= \lim_{n\to\infty} \textstyle\sum_{b\in B} |\sum_{a\in A} \vec{\mu}[n](a,b) - \sum_{a\in A} \mu(a,b)| \\
&= \lim_{n\to\infty} \textstyle\sum_{b\in B} |\sum_{a\in A} (\vec{\mu}[n](a,b) - \mu(a,b))| \\
&\leq \lim_{n\to\infty} \textstyle\sum_{b\in B} \sum_{a\in A} |\vec{\mu}[n](a,b) - \mu(a,b)| \\
&= \lim_{n\to\infty} \textstyle\sum_{a\in A} \sum_{b\in B} |\vec{\mu}[n](a,b) - \mu(a,b)| \qquad \text{(by Tonelli's Theorem)} \\
&= 0.
\end{aligned}$$

From $\sum_{b\in A}|\vec{\mu}[n]^{(B)}(b)-\mu^{(B)}(b)|\geq 0$ for all n we know $\lim_{n\to\infty}\sum_{b\in B}|\vec{\mu}[n]^{(B)}(b)-\mu^{(B)}(b)|\geq 0$, thus $\lim_{n\to\infty}\sum_{b\in B}|\vec{\mu}[n]^{(B)}(b)-\mu^{(B)}(b)|=0$. Therefore $\lim\vec{\mu}^{(B)}=\mu^{(B)}$.

Lemma 9. *Let A be any set, $\vec{\mu}$ be any infinite sequence of sub-distributions over A, μ be any sub-distribution over A, and $E\in A\to Prop$, if $\lim\vec{\mu}=\mu$, $\boldsymbol{Pr}_{a\sim\vec{\mu}[n]}[E(a)]>0$ for all n, $\vec{\mu}[n+1](a)\geq\vec{\mu}[n](a)$ for all n and a such that $E(a)$ holds, and $\lim_{n\to\infty}\boldsymbol{Pr}_{a\sim\vec{\mu}[n]}[E(a)]=1$, then $\lim\vec{\mu}|_E=\mu|_E$.*

Proof. From $\lim\vec{\mu}=\mu$ we know $\lim_{n\to\infty}\sum_{a\in A}|\vec{\mu}[n](a)-\mu(a)|=0$, and by Lem. 6 we know $\mu(a)=\lim_{n\to\infty}\vec{\mu}[n](a)$ for all a. From $\vec{\mu}[n+1](a)\geq\vec{\mu}[n](a)$ for all n and a such that $E(a)$ holds by Monotone Convergence Theorem for Series we know $\sum_a\{\lim_{n\to\infty}\vec{\mu}[n](a)\mid E(a)\}=\lim_{n\to\infty}\sum_a\{\vec{\mu}[n](a)\mid E(a)\}$, thus

$$\begin{aligned}&\mathbf{Pr}_{a\sim\mu}[E(a)]\\=&\textstyle\sum_a\{\mu(a)\mid E(a)\}\\=&\textstyle\sum_a\{\lim_{n\to\infty}\vec{\mu}[n](a)\mid E(a)\}\\=&\textstyle\lim_{n\to\infty}\sum_a\{\vec{\mu}[n](a)\mid E(a)\}\\=&\textstyle\lim_{n\to\infty}\mathbf{Pr}_{a\sim\vec{\mu}[n]}[E(a)]\\=&\,1.\end{aligned}$$

For all a such that $E(a)$ holds, we have $\vec{\mu}[n]|_E(a)-\mu|_E(a)=\frac{\vec{\mu}[n](a)}{\mathbf{Pr}_{a\sim\vec{\mu}[n]}[E(a)]}-\frac{\mu(a)}{\mathbf{Pr}_{a\sim\mu}[E(a)]}=\frac{\vec{\mu}[n](a)}{\mathbf{Pr}_{a\sim\vec{\mu}[n]}[E(a)]}-\mu(a)$, so $\vec{\mu}[n]|_E(a)-\mu|_E(a)\geq\vec{\mu}[n](a)-\mu(a)$. From $\vec{\mu}[n+1](a)\geq\vec{\mu}[n](a)$ for all n and $\mu(a)=\lim_{n\to\infty}\vec{\mu}[n](a)$ by Monotone Convergence Theorem we know $\mu(a)\geq\vec{\mu}[n](a)$ for all n, so $\vec{\mu}[n]|_E(a)-\mu|_E(a)=\frac{\vec{\mu}[n](a)}{\mathbf{Pr}_{a\sim\vec{\mu}[n]}[E(a)]}-\mu(a)\leq\frac{\mu(a)}{\mathbf{Pr}_{a\sim\vec{\mu}[n]}[E(a)]}-\mu(a)$. From $\vec{\mu}[n](a)-\mu(a)\leq\vec{\mu}[n]|_E(a)-\mu|_E(a)\leq\frac{\mu(a)}{\mathbf{Pr}_{a\sim\vec{\mu}[n]}[E(a)]}-\mu(a)$ we know $|\vec{\mu}[n]|_E(a)-\mu|_E(a)|\leq\max(|\vec{\mu}[n](a)-\mu(a)|,|\frac{\mu(a)}{\mathbf{Pr}_{a\sim\vec{\mu}[n]}[E(a)]}-\mu(a)|)$. From $\lim_{n\to\infty}\sum_a\{|\frac{\mu(a)}{\mathbf{Pr}_{a\sim\vec{\mu}[n]}[E(a)]}-\mu(a)|\mid E(a)\}=\sum_a\{\mu(a)\mid E(a)\}\cdot(\frac{1}{\lim_{n\to\infty}\mathbf{Pr}_{a\sim\vec{\mu}[n]}[E(a)]}-1)=0$ and $\lim_{n\to\infty}\sum_{a\in A}\{|\vec{\mu}[n](a)-\mu(a)|\mid E(a)\}\leq\lim_{n\to\infty}\sum_{a\in A}|\vec{\mu}[n](a)-\mu(a)|=0$ we know $\lim_{n\to\infty}\sum_{a\in A}\{\max(|\vec{\mu}[n](a)-\mu(a)|,|\frac{\mu(a)}{\mathbf{Pr}_{a\sim\vec{\mu}[n]}[E(a)]}-\mu(a)|)\mid E(a)\}=0$, thus $\lim_{n\to\infty}\sum_{a\in A}|\vec{\mu}[n]|_E(a)-\mu|_E(a)|=\lim_{n\to\infty}\sum_{a\in A}\{|\vec{\mu}[n]|_E(a)-\mu|_E(a)|\mid E(a)\}=0$. Therefore $\lim\vec{\mu}|_E=\mu|_E$.

Lemma 10. *Let A be any set, $\vec{\mu}$ be any infinite sequence of sub-distributions over A, μ be any sub-distribution over A, and N be any natural number, if $\lim\vec{\mu}=\mu$, then $\lim(\lambda n.\ \vec{\mu}[n+N])=\mu$.*

Proof. From $\lim \overrightarrow{\mu} = \mu$ we know $\lim_{n\to\infty} \sum_{a\in A} |\overrightarrow{\mu}[n](a) - \mu(a)| = 0$, thus $\lim_{n\to\infty} \sum_{a\in A} |\overrightarrow{\mu}[n+N](a) - \mu(a)| = 0$, so $\lim(\lambda n.\ \overrightarrow{\mu}[n+N]) = \mu$.

Lemma 11. *For all set A, B and $\mu_1, \mu_2 \in \mathbb{D}_{A\times B}$, $p \in [0,1]$, $(\mu_1 \oplus_p \mu_2)^{(A)} = {\mu_1}^{(A)} \oplus_p {\mu_2}^{(A)}$.*

Proof. For all set A, B and $\mu_1, \mu_2 \in \mathbb{D}_{A\times B}$, $p \in [0,1]$,

$$\begin{aligned}
&(\mu_1 \oplus_p \mu_2)^{(A)} \\
&= \lambda a. \textstyle\sum_b (\mu_1 \oplus_p \mu_2)(a,b) \\
&= \lambda a. \textstyle\sum_b p \cdot \mu_1(a,b) + (1-p) \cdot \mu_2(a,b) \\
&= \lambda a.\ p \cdot \textstyle\sum_b \mu_1(a,b) + (1-p) \cdot \sum_b \mu_2(a,b) \\
&= \lambda a.\ p \cdot {\mu_1}^{(A)}(a) + (1-p) \cdot {\mu_2}^{(A)}(a) \\
&= {\mu_1}^{(A)} \oplus_p {\mu_2}^{(A)}.
\end{aligned}$$

Lemma 12. *For all set A, B and $\mu_1, \mu_2 \in \mathbb{D}_{A\times B}$, $p \in [0,1]$, $(\mu_1 \oplus_p \mu_2)^{(B)} = {\mu_1}^{(B)} \oplus_p {\mu_2}^{(B)}$.*

Proof. For all set A, B and $\mu_1, \mu_2 \in \mathbb{D}_{A\times B}$, $p \in [0,1]$,

$$\begin{aligned}
&(\mu_1 \oplus_p \mu_2)^{(B)} \\
&= \lambda b. \textstyle\sum_a (\mu_1 \oplus_p \mu_2)(a,b) \\
&= \lambda b. \textstyle\sum_a p \cdot \mu_1(a,b) + (1-p) \cdot \mu_2(a,b) \\
&= \lambda b.\ p \cdot \textstyle\sum_a \mu_1(a,b) + (1-p) \cdot \sum_a \mu_2(a,b) \\
&= \lambda b.\ p \cdot {\mu_1}^{(B)}(b) + (1-p) \cdot {\mu_2}^{(B)}(b) \\
&= {\mu_1}^{(B)} \oplus_p {\mu_2}^{(B)}.
\end{aligned}$$

Lemma 13. *For all set A, B and $a \in A$, $\mu \in \mathbb{D}_{A\times B}$, if $\mu^{(A)}(a) = 1$, then $\mu = \delta(a) \otimes \mu^{(B)}$.*

Proof. For all set A, B and $a \in A$, $\mu \in \mathbb{D}_{A\times B}$ such that $\mu^{(A)}(a) = 1$, from $\mu^{(A)}(a) = 1$ we know $\mu(a', b) = 0$ for all a' and b such that $a' \neq a$, thus $\delta(a) \otimes \mu^{(B)} = \lambda(a', b).\ \delta(a)(a') \cdot \mu^{(B)}(b) = \lambda(a', b).\ \delta(a)(a') \cdot \sum_{a''} \mu(a'', b) = \lambda(a', b).\ \delta(a)(a') \cdot \mu(a, b) = \lambda(a', b).\ \mu(a', b)$.

Lemma 14. *For all set A, B and $\mu_1, \mu_2 \in \mathbb{D}_A, \mu_3 \in \mathbb{D}_B, p \in [0,1]$, $(\mu_1 \oplus_p \mu_2) \otimes \mu_3 = (\mu_1 \otimes \mu_3) \oplus_p (\mu_2 \otimes \mu_3)$.*

Proof. For all set A, B and $\mu_1, \mu_2 \in \mathbb{D}_A, \mu_3 \in \mathbb{D}_B, p \in [0,1]$,

$$\begin{aligned}
&(\mu_1 \oplus_p \mu_2) \otimes \mu_3 \\
&= \lambda(a,b).\ (\mu_1 \oplus_p \mu_2)(a) \cdot \mu_3(b) \\
&= \lambda(a,b).\ (p \cdot \mu_1(a) + (1-p) \cdot \mu_2(a)) \cdot \mu_3(b) \\
&= \lambda(a,b).\ p \cdot \mu_1(a) \cdot \mu_3(b) + (1-p) \cdot \mu_2(a) \cdot \mu_3(b) \\
&= \lambda(a,b).\ p \cdot (\mu_1 \otimes \mu_3)(a,b) + (1-p) \cdot (\mu_2 \otimes \mu_3)(a,b) \\
&= (\mu_1 \otimes \mu_3) \oplus_p (\mu_2 \otimes \mu_3).
\end{aligned}$$

Lemma 15. *For all set A, B, C and $\mu \in \mathbb{D}_A, f \in A \to \mathbb{D}_B, g \in B \to \mathbb{D}_C$, $\mathbb{E}_{b\sim\mathbb{E}_{a\sim\mu}\{f(a)\}}\{g(b)\} = \mathbb{E}_{a\sim\mu}\{\mathbb{E}_{b\sim f(a)}\{g(b)\}\}$.*

Proof. For all set A, B, C and $\mu \in \mathbb{D}_A, f \in A \to \mathbb{D}_B, g \in B \to \mathbb{D}_C$,

$$\begin{aligned}
&\mathbb{E}_{b\sim\mathbb{E}_{a\sim\mu}\{f(a)\}}\{g(b)\} \\
&= \lambda c.\, \textstyle\sum_b \mathbb{E}_{a\sim\mu}\{f(a)\}(b) \cdot g(b)(c) \\
&= \lambda c.\, \textstyle\sum_b \sum_a \mu(a) \cdot f(a)(b) \cdot g(b)(c) \\
&= \lambda c.\, \textstyle\sum_a \mu(a) \cdot \sum_b \cdot f(a)(b) \cdot g(b)(c) \\
&= \lambda c.\, \textstyle\sum_a \mu(a) \cdot \mathbb{E}_{b\sim f(a)}\{g(b)\}(c) \\
&= \mathbb{E}_{a\sim\mu}\{\mathbb{E}_{b\sim f(a)}\{g(b)\}\}.
\end{aligned}$$

Lemma 16. *For all set A, B and $\mu_1, \mu_2 \in \mathbb{D}_A, f \in A \to \mathbb{D}_B, p \in [0, 1]$, $\mathbb{E}_{a\sim\mu_1\oplus_p\mu_2}\{f(a)\} = \mathbb{E}_{a\sim\mu_1}\{f(a)\} \oplus_p \mathbb{E}_{a\sim\mu_2}\{f(a)\}$.*

Proof. For all set A, B and $\mu_1, \mu_2 \in \mathbb{D}_A, f \in A \to \mathbb{D}_B, p \in [0, 1]$,

$$\begin{aligned}
&\mathbb{E}_{a\sim\mu_1\oplus_p\mu_2}\{f(a)\} \\
&= \lambda b.\, \textstyle\sum_a (\mu_1 \oplus_p \mu_2)(a) \cdot f(a)(b) \\
&= \lambda b.\, \textstyle\sum_a (p \cdot \mu_1(a) + (1-p) \cdot \mu_2(a)) \cdot f(a)(b) \\
&= \lambda b.\ p \cdot \textstyle\sum_a \mu_1(a) f(a)(b) + (1-p) \cdot \sum_a \mu_2(a) \cdot f(a)(b) \\
&= \lambda b.\ p \cdot \mathbb{E}_{a\sim\mu_1}\{f(a)\}(b) + (1-p) \cdot \mathbb{E}_{a\sim\mu_2}\{f(a)\}(b) \\
&= \mathbb{E}_{a\sim\mu_1}\{f(a)\} \oplus_p \mathbb{E}_{a\sim\mu_2}\{f(a)\}.
\end{aligned}$$

Lemma 17. *For all set A, B and $f \in A \to \mathbb{D}_B, a \in A$, $\mathbb{E}_{a'\sim\delta(a)} = f(a)$.*

Proof. For all set A, B and $f \in A \to \mathbb{D}_B, a \in A$, we have $\mathbb{E}_{a'\sim\delta(a)} = \lambda b.\, \sum_{a'} \delta(a)(a') \cdot f(a')(b) = \lambda b.\ f(a)(b) = f(a)$.

Lemma 18. *For all set A, B and $\mu_1 \in \mathbb{D}_A, \mu_2 \in \mathbb{D}_B$, $(\mu_1 \otimes \mu_2)^{(A)} = \mu_1$.*

Proof. For all set A, B and $\mu_1 \in \mathbb{D}_A, \mu_2 \in \mathbb{D}_B$, $(\mu_1 \otimes \mu_2)^{(A)} = \lambda a.\, \sum_b (\mu_1 \otimes \mu_2)(a, b) = \lambda a.\, \sum_b \mu_1(a) \cdot \mu_2(b) = \lambda a.\ \mu_1(a) \cdot \sum_b \mu_2(b) = \lambda a.\ \mu_1(a) = \mu_1$.

Lemma 19. *For all set A, B and $\mu_1 \in \mathbb{D}_A, \mu_2 \in \mathbb{D}_B$, $(\mu_1 \otimes \mu_2)^{(B)} = \mu_2$.*

Proof. For all set A, B and $\mu_1 \in \mathbb{D}_A, \mu_2 \in \mathbb{D}_B$, $(\mu_1 \otimes \mu_2)^{(B)} = \lambda b.\, \sum_a (\mu_1 \otimes \mu_2)(a, b) = \lambda b.\, \sum_a \mu_1(a) \cdot \mu_2(b) = \lambda b.\ \mu_2(b) \cdot \sum_a \mu_1(a) = \lambda b.\ \mu_2(b) = \mu_2$.

Lemma 20. *For all set A and $\mu, \mu' \in \mathbb{D}_A, E \in A \to Prop$, if $\mu|_E = \mu'$, then $supp(\mu') \subseteq supp(\mu)$.*

Proof. For all set A and $\mu, \mu' \in \mathbb{D}_A, E \in A \to$ Prop such that $\mu|_E = \mu'$, $supp(\mu') = \{a \mid \mu'(a) > 0\} = \{a \mid \mu|_E(a) > 0\} = \{a \mid E(a) \wedge \mu(a) > 0\} \subseteq \{a \mid \mu(a) > 0\} \subseteq supp(\mu)$.

Lemma 21. *For all set A, B and $\mu \in \mathbb{D}_{A\times B}$, $supp(\mu^{(A)}) = dom(supp(\mu))$.*

Proof. For all set A, B and $\mu \in \mathbb{D}_{A\times B}$,

$$\begin{aligned} & supp(\mu^{(A)}) \\ = & \{a \mid \mu^{(A)}(a) > 0\} \\ = & \{a \mid \sum_b \mu(a,b) > 0\} \\ = & \{a \mid \exists b.\ \mu(a,b) > 0\} \\ = & \{a \mid \exists b.\ (a,b) \in supp(\mu)\} \\ = & dom(supp(\mu)). \end{aligned}$$

Lemma 22. *For all set A, B and $\mu \in \mathbb{D}_{A\times B}$, $supp(\mu^{(B)}) = range(supp(\mu))$.*

Proof. For all set A, B and $\mu \in \mathbb{D}_{A\times B}$,

$$\begin{aligned} & supp(\mu^{(B)}) \\ = & \{b \mid \mu^{(B)}(b) > 0\} \\ = & \{b \mid \sum_a \mu(a,b) > 0\} \\ = & \{b \mid \exists a.\ \mu(a,b) > 0\} \\ = & \{b \mid \exists a.\ (a,b) \in supp(\mu)\} \\ = & range(supp(\mu)). \end{aligned}$$

Lemma 23. *For all set A, B and $\mu_1, \mu_2 \in \mathbb{D}_{A\times B}$, if $supp(\mu_1) \subseteq supp(\mu_2)$, then $supp({\mu_1}^{(A)}) \subseteq supp({\mu_2}^{(A)})$.*

Proof. For all set A, B and $\mu_1, \mu_2 \in \mathbb{D}_{A\times B}$ such that $supp(\mu_1) \subseteq supp(\mu_2)$, by Lem. 21 we know $supp({\mu_1}^{(A)}) = dom(supp(\mu_1)) \subseteq dom(supp(\mu_2) = supp({\mu_2}^{(A)})$.

Lemma 24. *For all set A, B and $\mu_1, \mu_2 \in \mathbb{D}_{A\times B}$, if $supp(\mu_1) \subseteq supp(\mu_2)$, then $supp({\mu_1}^{(B)}) \subseteq supp({\mu_2}^{(B)})$.*

Proof. For all set A, B and $\mu_1, \mu_2 \in \mathbb{D}_{A\times B}$ such that $supp(\mu_1) \subseteq supp(\mu_2)$, by Lem. 22 we know $supp({\mu_1}^{(B)}) = range(supp(\mu_1)) \subseteq range(supp(\mu_2) = supp({\mu_2}^{(B)})$.

Lemma 25. *For all set A and $\mu \in \mathbb{D}_A, a \in A$, if $\mu(a) = 1$ then $\mu = \delta(a)$.*

Proof. For all set A and $\mu \in \mathbb{D}_A, a \in A$ such that $\mu(a) = 1$, we have $1 = |\mu| = \sum_{a'} \mu(a') = \mu(a) + \sum_{a'}\{\mu(a') \mid a' \neq a\} \leq \mu(a) = 1$, thus $\sum_{a'}\{\mu(a') \mid a' \neq a\} = 0$, so $\mu(a') = 0$ for all $a' \neq a$. From $\mu(a) = 1$ we know $\mu = \delta(a)$.

Lemma 26. *For all set A and $\mu \in \mathbb{D}_A, a \in A$, $\mu = \delta(a)$ if and only if $supp(\mu) = \{a\}$.*

Proof. For all set A and $\mu \in \mathbb{D}_A, a \in A$, we prove the two directions respectively.

- if $\mu = \delta(a)$, we have $supp(\mu') \subseteq supp(\mu) = \{a' \mid \mu(a') > 0\} = \{a' \mid \delta(a)(a') > 0\} = \{a\}$.
- if $supp(\mu) = \{a\}$, we have $1 = |\mu| = \sum_{a'} \mu(a') = \sum_{a'}\{\mu(a') \mid a' \in supp(\mu')\} = \sum_{a'}\{\mu(a') \mid a' \in \{a\}\} = \mu(a)$. By Lem. 25 we know $\mu' = \delta(a)$.

Lemma 27. *For all set A and $\mu, \mu' \in \mathbb{D}_A, a \in A$, if $\mu = \delta(a)$ and $supp(\mu') \subseteq supp(\mu)$, then $\mu' = \delta(a)$.*

Proof. For all set A and $\mu, \mu' \in \mathbb{D}_A, a \in A$ such that $\mu = \delta(a)$ and $supp(\mu') \subseteq supp(\mu)$, by Lem. 26 we know $supp(\mu) = \{a\}$, thus $supp(\mu') \subseteq \{a\}$. It is obvious that $supp(\mu') \neq \emptyset$, thus $supp(\mu') = \{a\}$. By Lem. 26 we know $\mu' = \delta(a)$.

H.2 Proof of Theorem 1

Proof (Proof of Theorem 1). For any $P, \mathbb{C}, Q$ such that $\models_{\text{A}} \{P\}\mathbb{C}\{Q\}$ and $\mathbf{closed}(Q)$, by Lem. 92 we have $\models_{\text{A}} \{P\}\mathbf{RemoveSplit}(\mathbb{C})\{Q\}$. It is obvious that $\mathbf{Nosplit}(\mathbf{RemoveSplit}(\mathbb{C}))$, by Lem. 68 we have $\models \{P\}\mathbf{RemoveSplit}(\mathbb{C})\{Q\}$. From Lem. 75 we have $\models \{P\}\mathbb{C}\{Q\}$.

The remainder of this section gives the proofs of the lemmas used in the proof of Thereorem 1.

Definition 39. Given $W_0, \varphi, \vec{W}$ such that $\mathbf{History}(W_0, \varphi, \vec{W})$. We write $W_0 \Downarrow'_\varphi \mu$ if and only if $\lim_{n\to\infty} \vec{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$ and $\forall\sigma.\ \lim_{n\to\infty} \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = \mu(\sigma)$.

Definition 40. $\models_{\text{A}'} \{P\}\mathbb{C}\{Q\}$ iff for all μ, if $\mu \models P$, then for all φ and μ', if $\mathit{init}(\mathbb{C}, \mu) \Downarrow'_\varphi \mu'$, then $\mu' \models Q$.

Lemma 28. *For all $\vec{W}$, if $\lim_{n\to\infty} \vec{W}[n]^{(Prog)}(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}) = 1$, then $\lim_{n\to\infty} \sum_{\mathbb{C},\sigma}\{\vec{W}[n](\mathbb{C}, \sigma) \mid \mathbb{C} \neq \boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}\} = 0$.*

Proof. For all $\vec{W}$ such that $\lim_{n\to\infty} \vec{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$, we have $\lim_{n\to\infty} \sum_\sigma \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = 1$. From $\forall n.\ |\vec{W}[n]| = 1$ we know $\lim_{n\to\infty} |\vec{W}[n]| = 1$, so

$$
\begin{aligned}
1 &= \lim_{n\to\infty} |\vec{W}[n]| \\
&= \lim_{n\to\infty} \textstyle\sum_{\mathbb{C},\sigma} \vec{W}[n](\mathbb{C}, \sigma) \\
&= \lim_{n\to\infty} (\textstyle\sum_\sigma \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) + \sum_{\mathbb{C},\sigma}\{\vec{W}[n](\mathbb{C}, \sigma) \mid \mathbb{C} \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\}) \\
&= \lim_{n\to\infty} \textstyle\sum_\sigma \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) + \lim_{n\to\infty} \sum_{\mathbb{C},\sigma}\{\vec{W}[n](\mathbb{C}, \sigma) \mid \mathbb{C} \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\} \\
&= 1 + \lim_{n\to\infty} \textstyle\sum_{\mathbb{C},\sigma}\{\vec{W}[n](\mathbb{C}, \sigma) \mid \mathbb{C} \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\},
\end{aligned}
$$

so $\lim_{n\to\infty} \sum_{\mathbb{C},\sigma}\{\vec{W}[n](\mathbb{C}, \sigma) \mid \mathbb{C} \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\} = 0$.

Lemma 29. *For all $\vec{W}, W$, if $\lim \vec{W} = W$ and $W^{(Prog)}(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}) = 1$, then for all σ, $W^{(State)}(\sigma) = \lim_{n\to\infty} \vec{W}[n](\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}, \sigma)$.*

Proof. For all $\vec{W}, W$ such that $\lim \vec{W} = W$ and $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$, from $\lim \vec{W} = W$ by Lem. 7 we know $\lim \vec{W}^{(Prog)} = W^{(Prog)}$. By Lem. 6 we

know $\lim_{n\to\infty} \vec{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$. By Lem. 28 we know $\lim_{n\to\infty} \sum_{\mathbb{C},\sigma}\{\vec{W}[n](\mathbb{C},\sigma) \mid \mathbb{C} \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\} = 0$. It is obvious that for all σ_1 and n, $0 \leq \sum_{\mathbb{C}}\{\vec{W}[n](\mathbb{C},\sigma_1) \mid \mathbb{C} \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\} \leq \sum_{\mathbb{C},\sigma}\{\vec{W}[n](\mathbb{C},\sigma) \mid \mathbb{C} \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\}$, by Squeeze Theorem we have $0 \leq \lim_{n\to\infty} \sum_{\mathbb{C}}\{\vec{W}[n](\mathbb{C},\sigma_1) \mid \mathbb{C} \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\} \leq \lim_{n\to\infty} \sum_{\mathbb{C},\sigma}\{\vec{W}[n](\mathbb{C},\sigma) \mid \mathbb{C} \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\} = 0$, thus $\lim_{n\to\infty} \sum_{\mathbb{C}}\{\vec{W}[n](\mathbb{C},\sigma)\} = 0$ for all σ. From $\lim \vec{W} = W$ by Lem. 8 we know $\lim \vec{W}^{(State)} = W^{(State)}$. By Lem. 6 we know for all σ, $\lim_{n\to\infty} \vec{W}[n]^{(State)}(\sigma) = W^{(State)}(\sigma)$, so

$$
\begin{aligned}
& W^{(State)}(\sigma) \\
= & \lim_{n\to\infty} \vec{W}[n]^{(State)}(\sigma) \\
= & \lim_{n\to\infty} \sum_{\mathbb{C}} \vec{W}[n](\mathbb{C},\sigma) \\
= & \lim_{n\to\infty} (\vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip},\sigma) + \sum_{\mathbb{C}}\{\vec{W}[n](\mathbb{C},\sigma) \mid \mathbb{C} \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\}) \\
= & \lim_{n\to\infty} \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip},\sigma) + \lim_{n\to\infty} \sum_{\mathbb{C}}\{\vec{W}[n](\mathbb{C},\sigma) \mid \mathbb{C} \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\} \\
= & \lim_{n\to\infty} \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip},\sigma) + 0 \\
= & \lim_{n\to\infty} \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip},\sigma)
\end{aligned}
$$

holds for all σ.

Lemma 30. *For all $n, W, \varphi, \vec{W}$, if **History**$(W, \varphi, \vec{W})$, then $\vec{W}[n] \overset{\varphi[n]}{\hookrightarrow} \vec{W}[n+1]$.*

Proof. by induction on n.

- base case: $n = 0$.
  From $\mathbf{History}(W, \varphi, \vec{W})$ there exists $t, \varphi', W', \vec{W}'$ such that $\varphi = t :: \varphi'$, $\vec{W} = W :: \vec{W}'$, $W \overset{t}{\hookrightarrow} W'$ and $\mathbf{History}(W', \varphi', \vec{W}')$. $\vec{W}[0] = (W :: \vec{W}')[0] = W$.
  From $\mathbf{History}(W', \varphi', \vec{W}')$ by Lem. 50 we know $\vec{W}'[0] = W'$, so $\vec{W}[1] = (W :: \vec{W}')[1] = \vec{W}'[0] = W'$. $\varphi[0] = (t :: \varphi')[0] = t$. Therefore $\vec{W}[0] \overset{\varphi[0]}{\hookrightarrow} \vec{W}[1]$.
- inductive case: $n = k + 1$.
  IH: for all $W, \varphi, \vec{W}$, if $\mathbf{History}(W, \varphi, \vec{W})$ and $\mathbf{Nosplit}(W)$, then $\vec{W}[n] \overset{\varphi[k]}{\hookrightarrow} \vec{W}[k+1]$.
  From $\mathbf{History}(W, \varphi, \vec{W})$ there exists $t, \varphi', W', \vec{W}'$ such that $\varphi = t :: \varphi'$, $\vec{W} = W :: \vec{W}'$, $W \overset{t}{\hookrightarrow} W'$ and $\mathbf{History}(W', \varphi', \vec{W}')$. From $\mathbf{History}(W', \varphi', \vec{W}')$

by IH we know $\vec{W}'[k] \overset{\varphi'[k]}{\hookrightarrow} \vec{W}'[k+1]$. $\vec{W}[n] = (W :: \vec{W}')[k+1] = \vec{W}'[k]$. $\vec{W}[n+1] = (W :: \vec{W}')[k+2] = \vec{W}'[k+1]$. $\varphi[n] = (t :: \varphi')[k+1] = \varphi'[k]$. Therefore $\vec{W}[n] \overset{\varphi[n]}{\hookrightarrow} \vec{W}[n+1]$.

Lemma 31. *For all W, W', t, if $W \overset{t}{\rightsquigarrow} W'$, then for all σ, $W'(\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}, \sigma) \geq W(\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}, \sigma)$.*

Proof. For all W, W', t, σ such that $W \overset{t}{\rightsquigarrow} W'$,

$$
\begin{aligned}
& W'(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \\
&= \textstyle\sum_{\mathbb{C}_1, \sigma_1} \{W(\mathbb{C}_1, \sigma_1) \mid (\mathbb{C}_1, \sigma_1) \overset{p}{\underset{t}{\rightarrow}} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)\} \\
&\geq W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \cdot p, \text{where } (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \overset{p}{\underset{t}{\rightarrow}} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \\
&= W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \cdot p, \text{where } (\mathbf{skip}, \sigma) \overset{p}{\rightarrow} (\mathbf{skip}, \sigma) \\
&= W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma).
\end{aligned}
$$

Lemma 32. *For all W, W', t, if $W \overset{t}{\hookrightarrow} W'$, then for all σ, $W'(\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}, \sigma) \geq W(\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}, \sigma)$.*

Proof. For all W, W', t such that $W \overset{t}{\hookrightarrow} W'$, it is obvious that $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 0 \vee W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$, we prove the two cases respectively.

- case 1: $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 0$.
  From $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 0$ we know $\sum_\sigma W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = 0$, thus $W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = 0$ for all σ. Therefore $W'(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \geq W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$ for all σ.
- case 2: $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$.
  From $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$ by Lem. 38 we know $\mathit{nextsplit}(W) \supseteq \mathbf{split}(\text{true})$. From $W \overset{t}{\hookrightarrow} W'$ by Lem. 48 we know $W \overset{t}{\rightsquigarrow} W'$. By Lem. 31 we know for all σ, $W'(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \geq W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$.

Lemma 33. *For all $W, \varphi, \vec{W}$, if $\textbf{\textit{History}}(W, \varphi, \vec{W})$, then for all n and σ, $\vec{W}[n+1](\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}, \sigma) \geq \vec{W}[n](\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}, \sigma)$.*

Proof. For all $W, \varphi, \vec{W}, n, \sigma$ such that $\mathbf{History}(W, \varphi, \vec{W})$, by Lem. 30 we know $\vec{W}[n] \overset{\varphi[n]}{\hookrightarrow} \vec{W}[n+1]$. By Lem. 32 we have $\vec{W}[n+1](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \geq \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$.

Lemma 34. *For all $W, \varphi, \vec{W}$, if $\textbf{\textit{History}}(W, \varphi, \vec{W})$ and $\lim_{n\to\infty} \vec{W}[n]^{(Prog)}(\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}) = 1$, then for all $\mathbb{C}, \sigma$, $\lim_{n\to\infty} \vec{W}[n](\mathbb{C}, \sigma)$ exists.*

Proof. For all $W, \varphi, \vec{W}, \mathbb{C}, \sigma$ such that $\mathbf{History}(W, \varphi, \vec{W})$ and $\lim_{n\to\infty} \vec{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$, it is obvious that $\mathbb{C} = \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}$ or $\mathbb{C} \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}$, we prove the two cases respectively.

- $\mathbb{C} = \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}$.
  From $\mathbf{History}(W, \varphi, \vec{W})$ by Lem. 33 we know $\forall n.\ \vec{W}[n+1](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \geq \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$. It is obvious $\forall n. \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \leq 1$, by Monotone Convergence Theorem we know $\lim_{n\to\infty} \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$ exists.
- $\mathbb{C} \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}$.
  From $\lim_{n\to\infty} \vec{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$ by Lem. 28 we know $\lim_{n\to\infty} \sum_{\mathbb{C}_1,\sigma_1} \{\vec{W}[n](\mathbb{C}_1, \sigma_1) \mid \mathbb{C}_1 \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\} = 0$.
  It is obvious that for all n, $0 \leq \vec{W}[n](\mathbb{C}, \sigma) \leq \sum_{\mathbb{C}_1,\sigma_1} \{\vec{W}[n](\mathbb{C}_1, \sigma_1) \mid \mathbb{C}_1 \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\}$, by Squeeze Theorem we have $0 \leq \lim_{n\to\infty} \vec{W}[n](\mathbb{C}, \sigma) \leq \lim_{n\to\infty} \sum_{\mathbb{C}_1,\sigma_1} \{\vec{W}[n](\mathbb{C}_1, \sigma_1) \mid \mathbb{C}_1 \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\} = 0$, thus $\lim_{n\to\infty} \vec{W}[n](\mathbb{C}, \sigma) = 0$.

Lemma 35. *For all W, there exists W_1 and W_2 such that $W = W_1 \oplus_{W^{(Prog)}(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip})} W_2$, $W_1{}^{(Prog)}(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}) = 1$ and $W_2{}^{(Prog)}(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}) = 0$.*

Proof. For all W, let $W_1 \stackrel{\text{def}}{=} \lambda(\mathbb{C}, \sigma).\ \frac{\delta(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})(\mathbb{C}) \cdot W(\mathbb{C}, \sigma)}{W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})}$,
$W_2 \stackrel{\text{def}}{=} \lambda(\mathbb{C}, \sigma).\ \frac{(1 - \delta(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})(\mathbb{C})) \cdot W(\mathbb{C}, \sigma)}{1 - W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})}$, then

$$
\begin{aligned}
& W_1 \oplus_{W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})} W_2 \\
&= \lambda(\mathbb{C}, \sigma).\ W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \cdot W_1(\mathbb{C}, \sigma) + (1 - W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})) \cdot W_2(\mathbb{C}, \sigma) \\
&= \lambda(\mathbb{C}, \sigma).\ \delta(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})(\mathbb{C}) \cdot W(\mathbb{C}, \sigma) + (1 - \delta(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})(\mathbb{C})) \cdot W(\mathbb{C}, \sigma) \\
&= \lambda(\mathbb{C}, \sigma).\ W(\mathbb{C}, \sigma) \\
&= W.
\end{aligned}
$$

Lemma 36. *For all $W_1, W_2, p, t, W_1', W_2'$, if $W_1 \overset{t}{\rightsquigarrow} W_1'$ and $W_2 \overset{t}{\rightsquigarrow} W_2'$, then $W_1 \oplus_p W_2 \overset{t}{\rightsquigarrow} W_1' \oplus_p W_2'$.*

Proof. For all $W_1, W_2, p, t, W_1', W_2'$ such that $W_1 \overset{t}{\rightsquigarrow} W_1'$ and $W_2 \overset{t}{\rightsquigarrow} W_2'$, we know $W_1' = \lambda(\mathbb{C}', \sigma').\ \sum_{\mathbb{C},\sigma} \{W_1(\mathbb{C}, \sigma) \cdot p' \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p'} (\mathbb{C}', \sigma')\}$, $W_2' = \lambda(\mathbb{C}', \sigma').\ \sum_{\mathbb{C},\sigma} \{W_2(\mathbb{C}, \sigma) \cdot$

$p' \mid (\mathbb{C},\sigma) \xrightarrow[t]{p'} (\mathbb{C}',\sigma')\}$, so

$$\begin{aligned}
&W_1' \oplus_p W_2' \\
&= \lambda(\mathbb{C}',\sigma').\ p \cdot \textstyle\sum_{\mathbb{C},\sigma}\{W_1(\mathbb{C},\sigma)\cdot p' \mid (\mathbb{C},\sigma) \xrightarrow[t]{p'} (\mathbb{C}',\sigma')\} + \\
&\qquad (1-p)\cdot \textstyle\sum_{\mathbb{C},\sigma}\{W_2(\mathbb{C},\sigma)\cdot p' \mid (\mathbb{C},\sigma) \xrightarrow[t]{p'} (\mathbb{C}',\sigma')\} \\
&= \lambda(\mathbb{C}',\sigma').\ \textstyle\sum_{\mathbb{C},\sigma}\{(p\cdot W_1(\mathbb{C},\sigma) + (1-p)\cdot W_2(\mathbb{C},\sigma))\cdot p' \mid (\mathbb{C},\sigma) \xrightarrow[t]{p'} (\mathbb{C}',\sigma')\} \\
&= \lambda(\mathbb{C}',\sigma').\ \textstyle\sum_{\mathbb{C},\sigma}\{(W_1 \oplus_p W_2)(\mathbb{C},\sigma)\cdot p' \mid (\mathbb{C},\sigma) \xrightarrow[t]{p'} (\mathbb{C}',\sigma')\},
\end{aligned}$$

thus $W_1 \oplus_p W_2 \overset{t}{\rightsquigarrow} W_1' \oplus_p W_2'$.

Lemma 37. *For all W_1, W_2, p, t, W', if $W_1 \oplus_p W_2 \overset{t}{\rightsquigarrow} W'$, then there exists W_1' and W_2' such that $W' = W_1' \oplus_p W_2'$, $W_1 \overset{t}{\rightsquigarrow} W_1'$ and $W_2 \overset{t}{\rightsquigarrow} W_2'$.*

Proof. For all W_1, W_2, p, t, W' such that $W_1 \oplus_p W_2 \overset{t}{\rightsquigarrow} W'$, let $W_1' \overset{\text{def}}{=} \lambda(\mathbb{C}',\sigma').\ \sum_{\mathbb{C},\sigma}\{W_1(\mathbb{C},\sigma)\cdot p \mid (\mathbb{C},\sigma) \xrightarrow[t]{p} (\mathbb{C}',\sigma')\}$, $W_2' \overset{\text{def}}{=} \lambda(\mathbb{C}',\sigma').\ \sum_{\mathbb{C},\sigma}\{W_2(\mathbb{C},\sigma)\cdot p \mid (\mathbb{C},\sigma) \xrightarrow[t]{p} (\mathbb{C}',\sigma')\}$, then $W_1 \overset{t}{\rightsquigarrow} W_1'$ and $W_2 \overset{t}{\rightsquigarrow} W_2'$. By Lem. 36 we know $W_1 \oplus_p W_2 \overset{t}{\rightsquigarrow} W_1' \oplus_p W_2'$. From $W_1 \oplus_p W_2 \overset{t}{\rightsquigarrow} W'$ by Lem. 57 we know $W' = W_1' \oplus_p W_2'$.

Lemma 38. *For all W, if $W^{(Prog)}(\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}) > 0$, then $nextsplit(W,t) \supseteq \{\textbf{\textit{split}}(true)\}$ for all t.*

Proof. For all W such that $W^{(Prog)}(\textbf{skip} \parallel \cdots \parallel \textbf{skip}) > 0$, we know $\sum_\sigma W(\textbf{skip} \parallel \cdots \parallel \textbf{skip}, \sigma) > 0$, thus there exists σ_1 such that $W(\textbf{skip} \parallel \cdots \parallel \textbf{skip}, \sigma_1) > 0$.

$$\begin{aligned}
&nextsplit(W,t) \\
&= \{nextsplit(C_t) \mid (C_1 \parallel \cdots \parallel C_n, \sigma) \in supp(W)\} \\
&= \{nextsplit(C_t) \mid W(C_1 \parallel \cdots \parallel C_n, \sigma) > 0\} \\
&\supseteq \{nextsplit(\textbf{skip})\} \\
&= \{\textbf{split}(\text{true})\}
\end{aligned}$$

Lemma 39. *For all W, if $W^{(Prog)}(\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}) = 1$, then $W \overset{t}{\rightsquigarrow} W$ for all t.*

Proof. For all W such that $W^{(Prog)}(\textbf{skip} \parallel \cdots \parallel \textbf{skip}) = 1$, by Lem. 13 we know $W = \delta(\textbf{skip} \parallel \cdots \parallel \textbf{skip}) \otimes W^{(State)}$. For all t, we have $W \overset{t}{\rightsquigarrow}$

$\lambda(\mathbb{C}', \sigma').\sum_{\mathbb{C},\sigma}\{W(\mathbb{C},\sigma) \mid (\mathbb{C},\sigma) \xrightarrow[t]{p} (\mathbb{C}',\sigma')\}$.

$$
\begin{aligned}
&\lambda(\mathbb{C}', \sigma').\sum_{\mathbb{C},\sigma}\{W(\mathbb{C},\sigma)\cdot p \mid (\mathbb{C},\sigma) \xrightarrow[t]{p} (\mathbb{C}',\sigma')\}\\
=\ &\lambda(\mathbb{C}', \sigma').\sum_{\mathbb{C},\sigma}\{\delta(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})(\mathbb{C})\cdot W^{(State)}(\sigma)\cdot p \mid (\mathbb{C},\sigma) \xrightarrow[t]{p} (\mathbb{C}',\sigma')\}\\
=\ &\lambda(\mathbb{C}', \sigma').\sum_{\sigma}\{W^{(State)}(\sigma)\cdot p \mid (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \xrightarrow[t]{p} (\mathbb{C}',\sigma')\}\\
=\ &\lambda(\mathbb{C}', \sigma').\sum_{\sigma}\{W^{(State)}(\sigma)\cdot p \mid \mathbb{C}' = \mathbf{skip} \parallel \cdots \parallel \mathbf{skip} \wedge \sigma' = \sigma \wedge p = 1\}\\
=\ &\lambda(\mathbb{C}', \sigma').\delta(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})(\mathbb{C}')\cdot W^{(State)}(\sigma')\\
=\ &\delta(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \otimes W^{(State)}\\
=\ &W.
\end{aligned}
$$

so $W \overset{t}{\rightsquigarrow} W$.

Lemma 40. *For all* $n, W, \varphi, \vec{W}, W_1, W_2, p$ *such that* ***History***$(W, \varphi, \vec{W})$, $W = W_1 \oplus_p W_2$ *and* ${W_1}^{(Prog)}(\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}) = 1$, *there exists* W' *such that* $\vec{W}[n] = W_1 \oplus_p W'$.

Proof. by induction on n.

- base case: $n = 0$.
  For all $W, \varphi, \vec{W}, W_1, W_2, p$ such that $\mathbf{History}(W, \varphi, \vec{W})$, $W = W_1 \oplus_p W_2$ and ${W_1}^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$, from $\mathbf{History}(W, \varphi, \vec{W})$ by Lem. 50 we know $\vec{W}[0] = W = W_1 \oplus_p W_2$. Let $W_2' \overset{\text{def}}{=} W_2$, then $\vec{W}[0] = W_1 \oplus_p W'$.
- inductive case: $n = k + 1$.
  IH: For all $W, \varphi, \vec{W}, W_1, W_2, p$ such that $\mathbf{History}(W, \varphi, \vec{W})$, $W = W_1 \oplus_p W_2$ and ${W_1}^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$, there exists W' such that $\vec{W}[k] = W_1 \oplus_p W'$.
  For all $W, \varphi, \vec{W}, W_1, W_2, p$ such that $\mathbf{History}(W, \varphi, \vec{W})$, $W = W_1 \oplus_p W_2$ and ${W_1}^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$, from $\mathbf{History}(W, \varphi, \vec{W})$ there exists $t, \varphi_0, W_0, \vec{W}_0$ such that $\varphi = t :: \varphi_0$, $W \overset{t}{\hookrightarrow} W_0$, $\vec{W} = W :: \vec{W}_0$ and $\mathbf{History}(W_0, \varphi_0, \vec{W}_0)$, then $\vec{W}[n] = (W :: \vec{W}_0)[k+1] = \vec{W}_0[k]$. It is obvious that $p = 0 \vee p > 0$, we prove the three cases respectively.
  - case 1: $p = 0$.
    Let $W' \overset{\text{def}}{=} \vec{W}_0[k]$, then $\vec{W}[n] = \vec{W}_0[k] = W_1 \oplus_0 W'$.
  - case 2: $p > 0$.

$$
\begin{aligned}
&{W}^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})\\
=\ &(W_1 \oplus_p W_2)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})\\
=\ &({W_1}^{(Prog)} \oplus_p {W_2}^{(Prog)})(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \qquad \text{(by Lem. 11)}\\
=\ &p \cdot {W_1}^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) + (1-p)\cdot {W_2}^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})\\
\geq\ &p > 0,
\end{aligned}
$$

by Lem. 38 we know $\mathit{nextsplit}(W,t) \supseteq \{\mathbf{split}(\mathrm{true})\}$. From $W \overset{t}{\hookrightarrow} W_0$ by Lem. 48 we know $W \overset{t}{\rightsquigarrow} W_0$. From $W = W_1 \oplus_p W_2$ by Lem. 36 there exists W_{01} and W_{02} such that $W_0 = W_{01} \oplus_p W_{02}$, $W_1 \overset{t}{\rightsquigarrow} W_{01}$ and $W_2 \overset{t}{\rightsquigarrow} W_{02}$. From ${W_1}^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$ by Lem. 39 we know $W_1 \overset{t}{\rightsquigarrow} W_1$. From $W_1 \overset{t}{\rightsquigarrow} W_{01}$ by Lem. 57 we have $W_1 = W_{01}$, so $W_0 = W_1 \oplus_p W_{02}$. From $\mathbf{History}(W_0, \varphi_0, \vec{W}_0)$, $W_0 = W_1 \oplus_p W_{02}$ and ${W_1}^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$ by IH there exists W' such that $\vec{W}_0[k] = W_1 \oplus_p W'$, thus $\vec{W}[n] = \vec{W}_0[k] = W_1 \oplus_p W'$.

Lemma 41. *For all* $W, \varphi, \vec{W}, W'$, *if* $\boldsymbol{History}(W, \varphi, \vec{W})$ *and* $\forall \mathbb{C}, \sigma. \lim_{n\to\infty} \vec{W}[n](\mathbb{C},\sigma) = W'(\mathbb{C},\sigma)$, *then* $\sum_{\mathbb{C},\sigma} |W(\mathbb{C},\sigma) - W'(\mathbb{C},\sigma)| \le 2(1 - {W}^{(Prog)}(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}))$.

Proof. For all $W, \varphi, \vec{W}, W'$ such that $\mathbf{History}(W, \varphi, \vec{W})$ and $\forall \mathbb{C}, \sigma. \lim_{n\to\infty} \vec{W}[n](\mathbb{C},\sigma) = W'(\mathbb{C},\sigma)$, let $p \overset{\text{def}}{=} {W}^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})$, by Lem. 35 there exists W_1 and W_2 such that $W = W_1 \oplus_p W_2$, ${W_1}^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$ and ${W_2}^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 0$. By Lem. 40 we know for all n, there exists W' such that $\vec{W}[n] = W_1 \oplus_p W'$. By Axiom of Choice, there exists $\vec{W}'$ such that for all n, $\vec{W}[n] = W_1 \oplus_p \vec{W}'[n]$. For all $\mathbb{C}$ and σ, $W'(\mathbb{C},\sigma) = \lim_{n\to\infty} \vec{W}[n](\mathbb{C},\sigma) = \lim_{n\to\infty} (p \cdot W_1(\mathbb{C},\sigma) + (1-p) \cdot \vec{W}'[n](\mathbb{C},\sigma)) = p \cdot W_1(\mathbb{C},\sigma) + (1-p) \cdot \lim_{n\to\infty} \vec{W}'[n](\mathbb{C},\sigma)$. Therefore,

$$
\begin{aligned}
&\textstyle\sum_{\mathbb{C},\sigma} |W(\mathbb{C},\sigma) - W'(\mathbb{C},\sigma)| \le 2(1 - {W}^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})) \\
&= \textstyle\sum_{\mathbb{C},\sigma} |p \cdot W_1(\mathbb{C},\sigma) + (1-p) \cdot W_2(\mathbb{C},\sigma) - p \cdot W_1(\mathbb{C},\sigma) - (1-p) \cdot \lim_{n\to\infty} \vec{W}'[n](\mathbb{C},\sigma)| \\
&= (1-p) \cdot \textstyle\sum_{\mathbb{C},\sigma} |W_2(\mathbb{C},\sigma) - \lim_{n\to\infty} \vec{W}'[n](\mathbb{C},\sigma)| \\
&\le (1-p) \cdot \textstyle\sum_{\mathbb{C},\sigma} (W_2(\mathbb{C},\sigma) + \lim_{n\to\infty} \vec{W}'[n](\mathbb{C},\sigma)) \\
&= (1-p) \cdot (\textstyle\sum_{\mathbb{C},\sigma} W_2(\mathbb{C},\sigma) + \sum_{\mathbb{C},\sigma} \lim_{n\to\infty} \vec{W}'[n](\mathbb{C},\sigma)) \\
&\le (1-p) \cdot (\textstyle\sum_{\mathbb{C},\sigma} W_2(\mathbb{C},\sigma) + \lim_{n\to\infty} \sum_{\mathbb{C},\sigma} \vec{W}'[n](\mathbb{C},\sigma)) \qquad \text{(by Fatou's Lemma)} \\
&= 2(1-p) \\
&= 2(1 - {W}^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})).
\end{aligned}
$$

Lemma 42. *For all* $n, W, \varphi, \vec{W}, W'$, *if* $\boldsymbol{History}(W, \varphi, \vec{W})$ *and* $\forall \mathbb{C}, \sigma. \lim_{n\to\infty} \vec{W}[n](\mathbb{C},\sigma) = W'(\mathbb{C},\sigma)$, *then* $\sum_{\mathbb{C},\sigma} |\vec{W}[n](\mathbb{C},\sigma) - W'(\mathbb{C},\sigma)| \le 2(1 - \vec{W}[n]^{(Prog)}(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}))$.

Proof. by induction on n.

- base case: $n = 0$.
  For all $W, \varphi, \vec{W}, W'$ such that $\mathbf{History}(W, \varphi, \vec{W})$ and $\forall \mathbb{C}, \sigma.\ \lim_{n\to\infty} \vec{W}[n](\mathbb{C}, \sigma) = W'(\mathbb{C}, \sigma)$, by Lem. 41 we have $\sum_{\mathbb{C},\sigma} |W(\mathbb{C}, \sigma) - W'(\mathbb{C}, \sigma)| \leq 2(1 - W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}))$. From $\mathbf{History}(W, \varphi, \vec{W})$ by Lem. 50 we know $\vec{W}[0] = W$, so $\sum_{\mathbb{C},\sigma} |\vec{W}[0](\mathbb{C}, \sigma) - W'(\mathbb{C}, \sigma)| \leq 2(1 - \vec{W}[0]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}))$.
- inductive: $n = k + 1$.
  IH: for all $W, \varphi, \vec{W}, W'$, if $\mathbf{History}(W, \varphi, \vec{W})$ and $\forall \mathbb{C}, \sigma.\ \lim_{n\to\infty} \vec{W}[n](\mathbb{C}, \sigma) = W'(\mathbb{C}, \sigma)$, then $\sum_{\mathbb{C},\sigma} |\vec{W}[k](\mathbb{C}, \sigma) - W'(\mathbb{C}, \sigma)| \leq 2(1 - \vec{W}[k]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}))$.
  For all $W, \varphi, \vec{W}, W'$ such that $\mathbf{History}(W, \varphi, \vec{W})$ and $\forall \mathbb{C}, \sigma.\ \lim_{n\to\infty} \vec{W}[n](\mathbb{C}, \sigma) = W'(\mathbb{C}, \sigma)$, from $\mathbf{History}(W, \varphi, \vec{W})$ there exists $t, \varphi', W_1, \vec{W}_1$ such that $\varphi = t :: \varphi'$, $W \xhookrightarrow{t} W_1$, $\vec{W} = W :: \vec{W}_1$ and $\mathbf{History}(W_1, \varphi', \vec{W}_1)$. For all $\mathbb{C}, \sigma$, $\lim_{n\to\infty} \vec{W}_1[n](\mathbb{C}, \sigma) = \lim_{n\to\infty} (W :: \vec{W}_1)[n+1](\mathbb{C}, \sigma) = \lim_{n\to\infty} \vec{W}[n+1](\mathbb{C}, \sigma) = \lim_{n\to\infty} \vec{W}[n](\mathbb{C}, \sigma) = W'(\mathbb{C}, \sigma)$. By IH we have $\sum_{\mathbb{C},\sigma} |\vec{W}_1[k](\mathbb{C}, \sigma) - W'(\mathbb{C}, \sigma)| \leq 2(1 - \vec{W}_1[k]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}))$. From $\vec{W}[n] = (W :: \vec{W}_1)[k+1] = \vec{W}_1[k]$ we know $\sum_{\mathbb{C},\sigma} |\vec{W}[n](\mathbb{C}, \sigma) - W'(\mathbb{C}, \sigma)| \leq 2(1 - \vec{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}))$.

Lemma 43. *For all* $W, \varphi, \vec{W}, W'$, *if* $\boldsymbol{History}(W, \varphi, \vec{W})$, $\lim_{n\to\infty} \vec{W}[n]^{(Prog)}(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}) = 1$, *and* $\forall \mathbb{C}, \sigma.\ \lim_{n\to\infty} \vec{W}[n](\mathbb{C}, \sigma) = W'(\mathbb{C}, \sigma)$, *then* $\lim \vec{W} = W'$.

Proof. For all $W, \varphi, \vec{W}, W'$ such that $\mathbf{History}(W, \varphi, \vec{W})$, $\lim_{n\to\infty} \vec{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$, and $\forall \mathbb{C}, \sigma.\ \lim_{n\to\infty} \vec{W}[n](\mathbb{C}, \sigma) = W'(\mathbb{C}, \sigma)$, by Lem. 42 we know for all n, $0 \leq \sum_{\mathbb{C},\sigma} |\vec{W}[n](\mathbb{C}, \sigma) - W'(\mathbb{C}, \sigma)| \leq 2(1 - \vec{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}))$. By Squeeze Theorem we have
$0 \leq \lim_{n\to\infty} \sum_{\mathbb{C},\sigma} |\vec{W}[n](\mathbb{C}, \sigma) - W'(\mathbb{C}, \sigma)| \leq \lim_{n\to\infty} 2(1 - \vec{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})) = 2(1 - \lim_{n\to\infty} \vec{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})) = 0$, so $\lim_{n\to\infty} \sum_{\mathbb{C},\sigma} |\vec{W}[n](\mathbb{C}, \sigma) - W'(\mathbb{C}, \sigma)| = 0$, by Def. 6 we have $\lim \vec{W} = W'$.

Lemma 44. *For all* $P, \mathbb{C}, Q$, $\models_{A} \{P\}\mathbb{C}\{Q\}$ *if and only if* $\models_{A'} \{P\}\mathbb{C}\{Q\}$.

Proof. For all $P, \mathbb{C}, Q$, first we prove if $\models_{A'} \{P\}\mathbb{C}\{Q\}$ then $\models_{A} \{P\}\mathbb{C}\{Q\}$. By definition of $\models_{A}$, we need to prove for all μ, φ, W, if $\mu \models P$ and $init(\mathbb{C}, \mu) \Downarrow_{\varphi} W$, then $W^{(State)} \models Q$. From $init(\mathbb{C}, \mu) \Downarrow_{\varphi} W$ we know there exists $\vec{W}$ such that

$\mathbf{History}(\mathit{init}(\mathbb{C},\mu),\varphi,\overrightarrow{W})$, $\lim\overrightarrow{W} = W$ and $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$. From $\lim\overrightarrow{W} = W$ by Lem. 7 we know $\lim\overrightarrow{W}^{(Prog)} = W^{(Prog)}$. By Lem. 6 we know $\lim_{n\to\infty}\overrightarrow{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$. From $\lim\overrightarrow{W} = W$ and $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$ by Lem. 29 we know for all σ, $W^{(State)}(\sigma) = \lim_{n\to\infty}\overrightarrow{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip},\sigma)$, thus $\mathit{init}(\mathbb{C},\sigma) \Downarrow'_{\varphi} W^{(State)}$. From $\models_{A'} \{P\}\mathbb{C}\{Q\}$, $\mu \models P$ we know $W^{(State)} \models Q$.

Then we prove if $\models_{A} \{P\}\mathbb{C}\{Q\}$ then $\models_{A'} \{P\}\mathbb{C}\{Q\}$. By definition of $\models_{A'}$, we need to prove for all μ,φ,μ', if $\mu \models P$ and $\mathit{init}(\mathbb{C},\mu) \Downarrow'_{\varphi} \mu'$, then $\mu' \models Q$. From $\mathit{init}(\mathbb{C},\mu) \Downarrow'_{\varphi} \mu'$ we know there exists $\overrightarrow{W}$ such that $\mathbf{History}(\mathit{init}(\mathbb{C},\mu),\varphi,\overrightarrow{W})$, $\lim_{n\to\infty}\overrightarrow{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$ and $\forall\sigma.\ \lim_{n\to\infty}\overrightarrow{W}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip},\sigma) = \mu'(\sigma)$. From $\mathbf{History}(\mathit{init}(\mathbb{C},\mu),\varphi,\overrightarrow{W})$ and $\lim_{n\to\infty}\overrightarrow{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$ by Lem. 34 we know $\lim_{n\to\infty}\overrightarrow{W}[n](\mathbb{C},\sigma)$ exists for all $\mathbb{C}$ and σ. Let $W \stackrel{\text{def}}{=} \lambda(\mathbb{C},\sigma).\ \lim_{n\to\infty}\overrightarrow{W}[n](\mathbb{C},\sigma)$, by Lem. 43 we know $\lim\overrightarrow{W} = W$. By Lem. 7 we know $\lim\overrightarrow{W}^{(Prog)} = W^{(Prog)}$. By Lem. 6 we know $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = \lim_{n\to\infty}\overrightarrow{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$. From $\mathbf{History}(\mathit{init}(\mathbb{C},\mu),\varphi,\overrightarrow{W})$, $\lim\overrightarrow{W} = W$ and $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$ we know $\mathit{init}(\mathbb{C},\mu) \Downarrow_{\varphi} W$. From $\models_{A} \{P\}\mathbb{C}\{Q\}$ and $\mu \models P$ we have $W^{(State)} \models Q$. From $\lim\overrightarrow{W} = W$ and $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$ by Lem. 29 we know $\forall\sigma.\ W^{(State)} = \lim_{n\to\infty}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip},\sigma)$. From $\forall\sigma.\ \lim_{n\to\infty}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip},\sigma) = \mu'(\sigma)$ we know $W^{(State)} = \mu'$. From $W^{(State)} \models Q$ we have $\mu' \models Q$.

Definition 41. $\mathbf{Nosplit}(C_1 \parallel \cdots \parallel C_n)$ if and only if $\mathbf{Nosplit}(C_1) \wedge \cdots \wedge \mathbf{Nosplit}(C_n)$.

Definition 42. $\mathbf{Nosplit}(W)$ if and only if $\forall(\mathbb{C},\sigma) \in \mathit{supp}(W).\ \mathbf{Nosplit}(\mathbb{C})$.

Lemma 45. *For all C, if* $\mathbf{Nosplit}(C)$ *then* $\mathit{nextsplit}(C) = \boldsymbol{split}(\mathit{true})$.

Proof. by induction on the structure of C.

- case 1: $C = \langle C_1\rangle$ *sp*, which contradicts with $\mathbf{Nosplit}(C)$.
- case 2: $C = C_1; C_2$. IH: if $\mathbf{Nosplit}(C_1)$ then $\mathit{nextsplit}(C_1) = \mathbf{split}(\mathrm{true})$. From $\mathbf{Nosplit}(C)$ we know $\mathbf{Nosplit}(C_1)$, so $\mathit{nextsplit}(C) = \mathit{nextsplit}(C_1; C_2) = \mathit{nextsplit}(C_1) = \mathbf{split}(\mathrm{true})$.
- other cases. $\mathit{nextsplit}(C) = \mathbf{split}(\mathrm{true})$.

Lemma 46. *For all W, t, if* $\mathbf{Nosplit}(W)$ *then* $\mathit{nextsplit}(W,t) = \{\boldsymbol{split}(\mathit{true})\}$.

Proof. For all W and t such that $\mathbf{Nosplit}(W)$, we have

$$
\begin{aligned}
&\mathit{nextsplit}(W,t)\\
&= \{\mathit{nextsplit}(C_t) \mid (C_1 \parallel \cdots \parallel C_n, \sigma) \in \mathit{supp}(W)\}\\
&= \{\mathit{nextsplit}(C_t) \mid (C_1 \parallel \cdots \parallel C_n, \sigma) \in \mathit{supp}(W) \wedge \mathbf{Nosplit}(C_1 \parallel \cdots \parallel C_n)\} && \text{(by Def. 42)}\\
&= \{\mathit{nextsplit}(C_t) \mid (C_1 \parallel \cdots \parallel C_n, \sigma) \in \mathit{supp}(W) \wedge \mathbf{Nosplit}(C_1) \wedge \cdots \wedge \mathbf{Nosplit}(C_n)\} && \text{(by Def. 41)}\\
&= \{\mathbf{split}(\text{true}) \mid (C_1 \parallel \cdots \parallel C_n, \sigma) \in \mathit{supp}(W) \wedge \mathbf{Nosplit}(C_1) \wedge \cdots \wedge \mathbf{Nosplit}(C_n)\} && \text{(by Lem. 45)}\\
&= \{\mathbf{split}(\text{true})\}
\end{aligned}
$$

Lemma 47. *For all W, $W|_{true} = W$.*

Proof. For all W, $W|_{\text{true}} = W|_{\lambda(\mathbb{C},\sigma).\sigma \models \text{true}} = W|_{\lambda(\mathbb{C},\sigma).\text{true}} = W$. The last step is by Lem. 5.

Lemma 48. *For all W, t, W', if $\mathit{nextsplit}(W) \supseteq \{\mathbf{split}(true)\}$, then $W \overset{t}{\rightsquigarrow} W'$ if and only if $W \overset{t}{\hookrightarrow} W'$.*

Proof. For all W, t, W' such that $\mathit{nextsplit}(W) \supseteq \{\mathbf{split}(\text{true})\}$, we prove the two directions respectively.

- if $W \overset{t}{\rightsquigarrow} W'$, from $\mathit{nextsplit}(W,t) \supseteq \{\mathbf{split}(\text{true})\}$ we know $\mathit{nextsplit}(W,t) = \{\mathbf{split}(\text{true})\}$ or $\mathit{nextsplit}(W,t) \supset \{\mathbf{split}(\text{true})\}$. We prove the two cases respectively.
  - $\mathit{nextsplit}(W,t) = \{\mathbf{split}(\text{true})\}$.
    By Lem. 47 we know $W'|_{\text{true}} = W'$. From $W \overset{t}{\rightsquigarrow} W'$, $\mathit{nextsplit}(W,t) = \{\mathbf{split}(\text{true})\}$ and $W'|_{\text{true}} = W'$ we have $W \overset{t}{\hookrightarrow} W'$.
  - $\mathit{nextsplit}(W,t) \supset \{\mathbf{split}(\text{true})\}$.
    $\#\mathit{nextsplit}(W,t) > 1$, so $W \overset{t}{\hookrightarrow} W'$.
- if $W \overset{t}{\hookrightarrow} W'$, there are two cases.
  - case 1: there exists $W'', b_1, \ldots, b_k, i$ such that $W \overset{t}{\rightsquigarrow} W''$, $\mathit{nextsplit}(W,t) = \{\mathbf{split}(b_1, \ldots, b_k)\}$ and $W''|_{b_i} = W'$.
    From $\mathit{nextsplit}(W,t) \supseteq \{\mathbf{split}(\text{true})\}$ we know $k = i = 1$, $b_1 = \text{true}$. By Lem. 47 we know $W''|_{\text{true}} = W''$, so $W' = W''|_{b_i} = W''|_{\text{true}} = W''$. From $W \overset{t}{\rightsquigarrow} W''$ we have $W \overset{t}{\rightsquigarrow} W'$.
  - case 2: $\#\mathit{nextsplit}(W) > 1$ and $W \overset{t}{\rightsquigarrow} W'$. trivial.

Lemma 49. *For all W, t, W', if $\mathbf{Nosplit}(W)$, then $W \overset{t}{\rightsquigarrow} W'$ if and only if $W \overset{t}{\hookrightarrow} W'$.*

Proof. For all W, t, W' such that $\mathbf{Nosplit}(W)$, by Lem. 46 we know $\mathit{nextsplit}(W) = \{\mathbf{split}(\text{true})\}$, so $\mathit{nextsplit}(W) \supseteq \{\mathbf{split}(\text{true})\}$. By Lem. 48 we know $W \overset{t}{\rightsquigarrow} W'$ if and only if $W \overset{t}{\hookrightarrow} W'$.

Lemma 50. *For all* $W, \varphi, \vec{W}$*, if* ***History***$(W, \varphi, \vec{W})$*, then* $\vec{W}[0] = W$.

Proof. For all $W, \varphi, \vec{W}$ such that **History**$(W, \varphi, \vec{W})$, there exists $t, \varphi', \vec{W}'$ such that $\varphi = t :: \varphi'$, $\vec{W} = W :: \vec{W}'$, $W \overset{t}{\hookrightarrow} W'$ and **History**$(W', \varphi', \vec{W}')$, so $\vec{W}[0] = (W :: \vec{W}')[0] = W$.

Lemma 51. *For all* $C, \sigma, C', \sigma', p$*, if* $(C, \sigma) \xrightarrow{p} (C', \sigma')$ *and* **Nosplit**(C)*, then* **Nosplit**(C').

Proof. by induction on the derivation of $(C, \sigma) \xrightarrow{p} (C', \sigma')$.

- case 1: $C = C' = \mathbf{skip}, \sigma = \sigma', p = 1$.
  From **Nosplit**(**skip**) we know **Nosplit**(C').
- case 2: $C = x := e, C' = \mathbf{skip}, \sigma' = \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}, p = 1$.
  From **Nosplit**(**skip**) we know **Nosplit**(C').
- case 3: $C = \mathbf{skip}; C_2, C' = C_2, \sigma = \sigma', p = 1$.
  From **Nosplit**(C) we know **Nosplit**(C_2).
- case 4: $C = C_1; C_2, C_1 \neq \mathbf{skip}, C' = C_1'; C_2, (C_1, \sigma) \xrightarrow{p} (C_1', \sigma')$.
  IH: if **Nosplit**(C_1) then **Nosplit**(C_1').
  From **Nosplit**(C) we know **Nosplit**(C_1) and **Nosplit**(C_2). By IH we have **Nosplit**(C_1'), so **Nosplit**$(C_1'; C_2)$, i.e., **Nosplit**(C').
- case 5: $C = \mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2, [\![b]\!]_\sigma = \mathrm{tt}, C' = C_1, \sigma' = \sigma, p = 1$.
  From **Nosplit**(C) we know **Nosplit**(C_1) and **Nosplit**(C_2), so **Nosplit**(C').
- case 6: $C = \mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2, [\![b]\!]_\sigma = \mathrm{ff}, C' = C_2, \sigma' = \sigma, p = 1$.
  From **Nosplit**(C) we know **Nosplit**(C_1) and **Nosplit**(C_2), so **Nosplit**(C').
- case 7: $C = \mathbf{while}\ (b)\ \mathbf{do}\ C_1, [\![b]\!]_\sigma = \mathrm{tt}, C' = C_1; \mathbf{while}\ (b)\ \mathbf{do}\ C_1, \sigma' = \sigma, p = 1$.
  From **Nosplit**(C) we know **Nosplit**(C_1), so **Nosplit**$(C_1; \mathbf{while}\ (b)\ \mathbf{do}\ C_1)$, i.e., **Nosplit**$(C')$.
- case 8: $C = \mathbf{while}\ (b)\ \mathbf{do}\ C_1, [\![b]\!]_\sigma = \mathrm{ff}, C' = \mathbf{skip}, \sigma' = \sigma, p = 1$.
  From **Nosplit**(**skip**) we know **Nosplit**(C').
- case 9: $C = \langle C_1 \rangle, C' = \mathbf{skip}$.
  From **Nosplit**(**skip**) we know **Nosplit**(C').
- case 10: $C = \langle C_1 \rangle\ sp, C' = \mathbf{skip}, (\langle C_1 \rangle, \sigma) \xrightarrow{p} (\mathbf{skip}, \sigma')$.
  $C = \langle C_1 \rangle\ sp$ contradicts with **Nosplit**(C).
- case 11: $C = \langle C_1 \rangle \oplus_{p'} \langle C_2 \rangle, C' = \langle C_1 \rangle, \sigma = \sigma', p = p'$.
  From **Nosplit**$(\langle C_1 \rangle)$ we know **Nosplit**(C').
- case 12: $C = \langle C_1 \rangle \oplus_{p'} \langle C_2 \rangle, C' = \langle C_2 \rangle, \sigma = \sigma', p = 1 - p'$.
  From **Nosplit**$(\langle C_2 \rangle)$ we know **Nosplit**(C').

Lemma 52. *For all* $\mathbb{C}, \sigma, \mathbb{C}', \sigma', t, p$*, if* $(\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')$ *and* **Nosplit**$(\mathbb{C})$*, then* **Nosplit**$(\mathbb{C}')$.

Proof. For all $\mathbb{C}, \sigma, \mathbb{C}', \sigma', t, p$ such that $(\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')$ and **Nosplit**$(\mathbb{C})$, there exists $C_1, \ldots, C_n, C_t'$ such that $\mathbb{C} = C_1 \parallel \cdots \parallel C_n$, $\mathbb{C}' = C_1 \parallel \ldots C_{t-1} \parallel$

$C'_t \parallel C_{t+1} \parallel \cdots \parallel C_n$, and $(C_t, \sigma) \xrightarrow{p} (C'_t, \sigma')$. From $\mathbf{Nosplit}(\mathbb{C})$ we know $\mathbf{Nosplit}(C_1) \wedge \cdots \wedge \mathbf{Nosplit}(C_n)$. From $\mathbf{Nosplit}(C_t)$ and $(C_t, \sigma) \xrightarrow{p} (C'_t, \sigma')$ by Lem. 51 we have $\mathbf{Nosplit}(C'_t)$.

Lemma 53. *For all* W, t, W', *if* $\mathbf{Nosplit}(W)$ *and* $W \overset{t}{\rightsquigarrow} W'$, *then* $\mathbf{Nosplit}(W')$.

Proof. For all W, t, W' such that $\mathbf{Nosplit}(W)$ and $W \overset{t}{\rightsquigarrow} W'$, we know $W' = \lambda(\mathbb{C}', \sigma').\sum_{\mathbb{C},\sigma}\{W(\mathbb{C}, \sigma) \cdot p \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')\}$. For all $(\mathbb{C}', \sigma') \in \mathit{supp}(W')$, we know $\sum_{\mathbb{C},\sigma}\{W(\mathbb{C}, \sigma) \cdot p \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')\} > 0$, so there exists W, σ such that $W(\mathbb{C}, \sigma) > 0 \wedge p > 0 \wedge (\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')$. From $W(\mathbb{C}, \sigma) > 0$ we know $(\mathbb{C}, \sigma) \in \mathit{supp}(W)$. From $\mathbf{Nosplit}(W)$ we know $\mathbf{Nosplit}(\mathbb{C})$. From $(\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')$ by lem. 52 we know $\mathbf{Nosplit}(\mathbb{C}')$.

Lemma 54. *For all* W, t, W', *if* $\mathbf{Nosplit}(W)$ *and* $W \overset{t}{\hookrightarrow} W'$, *then* $\mathbf{Nosplit}(W')$.

Proof. For all W, t, W' such that $\mathbf{Nosplit}(W)$ and $W \overset{t}{\hookrightarrow} W'$, by Lem. 49 we know $W \overset{t}{\rightsquigarrow} W'$, by Lem. 53 we have $\mathbf{Nosplit}(W')$.

Lemma 55. *For all* $n, W, \varphi, \overrightarrow{W}$, *if* $\boldsymbol{History}(W, \varphi, \overrightarrow{W})$ *and* $\mathbf{Nosplit}(W)$, *then* $\overrightarrow{W}[n] \overset{\varphi[n]}{\rightsquigarrow} \overrightarrow{W}[n+1]$.

Proof. by induction on n.

- base case: $n = 0$.
  From $\mathbf{History}(W, \varphi, \overrightarrow{W})$ there exists $t, \varphi', W', \overrightarrow{W}'$ such that $\varphi = t :: \varphi'$, $\overrightarrow{W} = W :: \overrightarrow{W}'$, $W \overset{t}{\hookrightarrow} W'$ and $\mathbf{History}(W', \varphi', \overrightarrow{W}')$. From $W \overset{t}{\hookrightarrow} W'$ by Lem. 49 we know $W \overset{t}{\rightsquigarrow} W'$. $\overrightarrow{W}[0] = (W :: \overrightarrow{W}')[0] = W$. From $\mathbf{History}(W', \varphi', \overrightarrow{W}')$ by Lem. 50 we know $\overrightarrow{W}'[0] = W'$, so $\overrightarrow{W}[1] = (W :: \overrightarrow{W}')[1] = \overrightarrow{W}'[0] = W'$. $\varphi[0] = (t :: \varphi')[0] = t$. Therefore $\overrightarrow{W}[0] \overset{\varphi[0]}{\rightsquigarrow} \overrightarrow{W}[1]$.
- inductive case: $n = k + 1$.
  IH: for all $W, \varphi, \overrightarrow{W}$, if $\mathbf{History}(W, \varphi, \overrightarrow{W})$ and $\mathbf{Nosplit}(W)$, then $\overrightarrow{W}[n] \overset{\varphi[k]}{\rightsquigarrow} \overrightarrow{W}[k+1]$.
  From $\mathbf{History}(W, \varphi, \overrightarrow{W})$ there exists $t, \varphi', W', \overrightarrow{W}'$ such that $\varphi = t :: \varphi'$, $\overrightarrow{W} = W :: \overrightarrow{W}'$, $W \overset{t}{\hookrightarrow} W'$ and $\mathbf{History}(W', \varphi', \overrightarrow{W}')$. From $\mathbf{Nosplit}(W)$ and $W \overset{t}{\hookrightarrow} W'$ by Lem. 54 we know $\mathbf{Nosplit}(W')$. From $\mathbf{History}(W', \varphi', \overrightarrow{W}')$ by IH we know $\overrightarrow{W}'[k] \overset{\varphi'[k]}{\rightsquigarrow} \overrightarrow{W}'[k+1]$. $\overrightarrow{W}[n] = (W :: \overrightarrow{W}')[k+1] = \overrightarrow{W}'[k]$.

$\vec{W}[n+1] = (W :: \vec{W}')[k+2] = \vec{W}'[k+1]$. $\varphi[n] = (t :: \varphi')[k+1] = \varphi'[k]$. Therefore $\vec{W}[n] \overset{\varphi[n]}{\rightsquigarrow} \vec{W}[n+1]$.

Lemma 56. *For all* $\mathbb{C}, \mu$, *if* **Nosplit**$(\mathbb{C})$, *then* **Nosplit**$(\mathit{init}(\mathbb{C}, \mu))$.

Proof. For all $\mathbb{C}, \mu$ such that **Nosplit**$(\mathbb{C})$, we need to prove $\forall (\mathbb{C}', \sigma) \in \mathit{supp}(\mathit{init}(\mathbb{C}, \mu))$. **Nosplit**$(\mathbb{C})$. For all $(\mathbb{C}', \sigma) \in \mathit{supp}(\mathit{init}(\mathbb{C}, \mu))$, we know $(\delta(\mathbb{C}) \otimes \mu)(\mathbb{C}', \sigma) > 0$, i.e., $\delta(\mathbb{C})(\mathbb{C}') \cdot \mu(\sigma) > 0$, so $\mathbb{C}' = \mathbb{C}$. From **Nosplit**$(\mathbb{C})$ we have **Nosplit**$(\mathbb{C}')$.

Lemma 57. *For all* W, W_1, W_2, t, *if* $W \overset{t}{\rightsquigarrow} W_1$ *and* $W \overset{t}{\rightsquigarrow} W_2$, *then* $W_1 = W_2$.

Proof. From $W \overset{t}{\rightsquigarrow} W_1$ we know $W_1 = \lambda(\mathbb{C}', \sigma').\ \sum_{\mathbb{C},\sigma}\{p \cdot W(\mathbb{C}, \sigma) \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')\}$. From $W \overset{t}{\rightsquigarrow} W_2$ we know $W_2 = \lambda(\mathbb{C}', \sigma').\ \sum_{\mathbb{C},\sigma}\{p \cdot W(\mathbb{C}, \sigma) \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')\}$. Therefore, $W_1 = W_2$.

Definition 43. $\mathit{Level}(\mathbb{C}, \sigma, \varphi, n) \overset{\text{def}}{=} \lambda(\mathbb{C}', \sigma').\ p$, where $(\mathbb{C}, \sigma) \xrightarrow[\varphi]{p}{}^{n} (\mathbb{C}', \sigma')$.

Definition 44. $\mathit{Level}(\mathbb{C}, \mu, \varphi, n) \overset{\text{def}}{=} \mathbb{E}_{\sigma\sim\mu}\{\mathit{Level}(\mathbb{C}, \sigma, \varphi, n)\}$

Lemma 58. *For all* $n, \mathbb{C}, \sigma, \varphi$, $\mathit{Level}(\mathbb{C}, \sigma, \varphi, n) \overset{\varphi[n]}{\rightsquigarrow} \mathit{Level}(\mathbb{C}, \sigma, \varphi, n+1)$.

Proof. by induction on n.

- base case: $n = 0$.

$$
\begin{aligned}
&\mathit{Level}(\mathbb{C}, \sigma, \varphi, 0)\\
&= \lambda(\mathbb{C}', \sigma').\ p, \text{where } (\mathbb{C}, \sigma) \xrightarrow[\varphi]{p}{}^{0} (\mathbb{C}', \sigma')\\
&= \lambda(\mathbb{C}', \sigma').\begin{cases} 1, & \text{if } \mathbb{C}' = \mathbb{C} \wedge \sigma' = \sigma \\ 0, & \text{otherwise} \end{cases}\\
&= \delta(\mathbb{C}, \sigma).
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{Level}(\mathbb{C}, \sigma, \varphi, 1)\\
&= \lambda(\mathbb{C}', \sigma').\ p, \text{where } (\mathbb{C}, \sigma) \xrightarrow[\varphi]{p}{}^{1} (\mathbb{C}', \sigma')\\
&= \lambda(\mathbb{C}', \sigma').\ p, \text{where } (\mathbb{C}, \sigma) \xrightarrow[\varphi[0]]{p} (\mathbb{C}', \sigma')\\
&= \lambda(\mathbb{C}', \sigma').\ \textstyle\sum_{\mathbb{C}_1,\sigma_1}\{\delta(\mathbb{C}, \sigma)(\mathbb{C}_1, \sigma_1) \cdot p \mid (\mathbb{C}, \sigma) \xrightarrow[\varphi[0]]{p} (\mathbb{C}', \sigma')\}\\
&= \lambda(\mathbb{C}', \sigma').\ \textstyle\sum_{\mathbb{C}_1,\sigma_1}\{\mathit{Level}(\mathbb{C}, \sigma, \varphi, 0)((\mathbb{C}_1, \sigma_1)) \cdot p \mid (\mathbb{C}, \sigma) \xrightarrow[\varphi[0]]{p} (\mathbb{C}', \sigma')\}.
\end{aligned}
$$

Therefore $\mathit{Level}(\mathbb{C}, \sigma, \varphi, 0) \overset{\varphi[0]}{\rightsquigarrow} \mathit{Level}(\mathbb{C}, \sigma, \varphi, 1)$.

– inductive case: $n = k + 1$.
IH: for all $\mathbb{C}, \sigma, \varphi$, $\mathit{Level}(\mathbb{C}, \sigma, \varphi, k) \overset{\varphi[k]}{\rightsquigarrow} \mathit{Level}(\mathbb{C}, \sigma, \varphi, k+1)$.
By IH we have for all $\mathbb{C}, \sigma, \varphi, \mathbb{C}', \sigma'$,
$\mathit{Level}(\mathbb{C}, \sigma, \varphi, k{+}1)(\mathbb{C}', \sigma') = \sum_{\mathbb{C}_1, \sigma_1} \{\mathit{Level}(\mathbb{C}, \sigma, \varphi, k)(\mathbb{C}_1, \sigma_1) \cdot p \mid (\mathbb{C}_1, \sigma_1) \xrightarrow[\varphi[k]]{p} (\mathbb{C}', \sigma')\}$.
By definition of *Schedule* there exists t and φ' such that $\varphi = t :: \varphi'$.

$$
\begin{aligned}
&\mathit{Level}(\mathbb{C}, \sigma, \varphi, n+1)\\
&= \mathit{Level}(\mathbb{C}, \sigma, t :: \varphi', k+2)\\
&= \lambda(\mathbb{C}', \sigma').\ p, \text{where } (\mathbb{C}, \sigma) \xrightarrow[t::\varphi']{p}{}^{k+2} (\mathbb{C}', \sigma')\\
&= \lambda(\mathbb{C}', \sigma').\ \textstyle\sum_{\mathbb{C}_1, \sigma_1} \{p_1 \cdot p_2 \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p_1} (\mathbb{C}_1, \sigma_1) \wedge (\mathbb{C}_1, \sigma_1) \xrightarrow[\varphi']{p_2}{}^{k+1} (\mathbb{C}', \sigma')\}\\
&= \lambda(\mathbb{C}', \sigma').\ \textstyle\sum_{\mathbb{C}_1, \sigma_1} \{p_1 \cdot \mathit{Level}(\mathbb{C}_1, \sigma_1, \varphi', k+1)(\mathbb{C}', \sigma') \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p_1} (\mathbb{C}_1, \sigma_1)\}\\
&= \lambda(\mathbb{C}', \sigma').\ \textstyle\sum_{\mathbb{C}_1, \sigma_1} \{p_1 \cdot \sum_{\mathbb{C}_1', \sigma_1'} \{\mathit{Level}(\mathbb{C}_1, \sigma_1, \varphi', k)(\mathbb{C}_1', \sigma_1') \cdot p \mid (\mathbb{C}_1', \sigma_1') \xrightarrow[\varphi'[k]]{p} (\mathbb{C}', \sigma')\} \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p_1} (\mathbb{C}_1, \sigma_1)\}\\
&= \lambda(\mathbb{C}', \sigma').\ \textstyle\sum_{\mathbb{C}_1', \sigma_1'} \{\sum_{\mathbb{C}_1, \sigma_1} \{p_1 \cdot \mathit{Level}(\mathbb{C}_1, \sigma_1, \varphi', k)(\mathbb{C}_1', \sigma_1') \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p_1} (\mathbb{C}_1, \sigma_1)\} \cdot p \mid (\mathbb{C}_1', \sigma_1') \xrightarrow[\varphi'[k]]{p} (\mathbb{C}', \sigma')\}\\
&= \lambda(\mathbb{C}', \sigma').\ \textstyle\sum_{\mathbb{C}_1', \sigma_1'} \{\sum_{\mathbb{C}_1, \sigma_1} \{p_1 \cdot p_2 \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p_1} (\mathbb{C}_1, \sigma_1) \wedge (\mathbb{C}_1, \sigma_1) \xrightarrow[\varphi']{p_2}{}^{k} (\mathbb{C}_1', \sigma_1')\} \cdot p \mid (\mathbb{C}_1', \sigma_1') \xrightarrow[\varphi'[k]]{p} (\mathbb{C}', \sigma')\}\\
&= \lambda(\mathbb{C}', \sigma').\ \textstyle\sum_{\mathbb{C}_1', \sigma_1'} \{p' \cdot p \mid (\mathbb{C}, \sigma) \xrightarrow[t::\varphi']{p'}{}^{k+1} (\mathbb{C}_1', \sigma_1') \wedge (\mathbb{C}_1', \sigma_1') \xrightarrow[\varphi'[k]]{p} (\mathbb{C}', \sigma')\}\\
&= \lambda(\mathbb{C}', \sigma').\ \textstyle\sum_{\mathbb{C}_1', \sigma_1'} \{\mathit{Level}(\mathbb{C}, \sigma, t :: \varphi', k+1)(\mathbb{C}_1', \sigma_1') \cdot p \mid (\mathbb{C}_1', \sigma_1') \xrightarrow[\varphi'[k]]{p} (\mathbb{C}', \sigma')\}\\
&= \lambda(\mathbb{C}', \sigma').\ \textstyle\sum_{\mathbb{C}_1', \sigma_1'} \{\mathit{Level}(\mathbb{C}, \sigma, \varphi, n)(\mathbb{C}_1', \sigma_1') \cdot p \mid (\mathbb{C}_1', \sigma_1') \xrightarrow[\varphi[n]]{p} (\mathbb{C}', \sigma')\}.
\end{aligned}
$$

Therefore $\mathit{Level}(\mathbb{C}, \sigma, \varphi, n) \overset{\varphi[n]}{\rightsquigarrow} \mathit{Level}(\mathbb{C}, \sigma, \varphi, n+1)$.

Lemma 59. *For all* $n, \mathbb{C}, \mu, \varphi$, $\mathit{Level}(\mathbb{C}, \mu, \varphi, n) \overset{\varphi[n]}{\rightsquigarrow} \mathit{Level}(\mathbb{C}, \mu, \varphi, n+1)$.

Proof. For all $n, \mathbb{C}, \varphi, \mu$, by Lem. 58 we know for all σ, $\mathit{Level}(\mathbb{C}, \sigma, \varphi, n) \overset{t}{\rightsquigarrow} \mathit{Level}(\mathbb{C}, \sigma, \varphi, n{+}1)$, so for all $\mathbb{C}', \sigma'$, $\mathit{Level}(\mathbb{C}, \sigma, \varphi, n{+}1)(\mathbb{C}', \sigma') = \sum_{\mathbb{C}_1, \sigma_1} \{\mathit{Level}(\mathbb{C}, \sigma, \varphi, n)(\mathbb{C}_1, \sigma_1) \cdot p \mid (\mathbb{C}_1, \sigma_1) \xrightarrow[\varphi[n]]{p} (\mathbb{C}', \sigma')\}$, thus

$$
\begin{aligned}
&\mathit{Level}(\mathbb{C}, \mu, \varphi, n+1)\\
&= \mathbb{E}_{\sigma \sim \mu}\{\mathit{Level}(\mathbb{C}, \sigma, \varphi, n+1)\}\\
&= \lambda(\mathbb{C}', \sigma').\textstyle\sum_{\sigma} \mu(\sigma) \cdot \mathit{Level}(\mathbb{C}, \sigma, \varphi, n+1)(\mathbb{C}', \sigma')\\
&= \lambda(\mathbb{C}', \sigma').\textstyle\sum_{\sigma} \mu(\sigma) \cdot \sum_{\mathbb{C}_1, \sigma_1} \{\mathit{Level}(\mathbb{C}, \sigma, \varphi, n)(\mathbb{C}_1, \sigma_1) \cdot p \mid (\mathbb{C}_1, \sigma_1) \xrightarrow[\varphi[n]]{p} (\mathbb{C}', \sigma')\}\\
&= \lambda(\mathbb{C}', \sigma').\textstyle\sum_{\mathbb{C}_1, \sigma_1} \{\sum_{\sigma} \mu(\sigma) \cdot \mathit{Level}(\mathbb{C}, \sigma, \varphi, n)(\mathbb{C}_1, \sigma_1) \cdot p \mid (\mathbb{C}_1, \sigma_1) \xrightarrow[\varphi[n]]{p} (\mathbb{C}', \sigma')\}\\
&= \lambda(\mathbb{C}', \sigma').\textstyle\sum_{\mathbb{C}_1, \sigma_1} \{\mathbb{E}_{\sigma \sim \mu}\{\mathit{Level}(\mathbb{C}, \sigma, \varphi, n)\}(\mathbb{C}_1, \sigma_1) \cdot p \mid (\mathbb{C}_1, \sigma_1) \xrightarrow[\varphi[n]]{p} (\mathbb{C}', \sigma')\}\\
&= \lambda(\mathbb{C}', \sigma').\textstyle\sum_{\mathbb{C}_1, \sigma_1} \{\mathit{Level}(\mathbb{C}, \mu, \varphi, n)(\mathbb{C}_1, \sigma_1) \cdot p \mid (\mathbb{C}_1, \sigma_1) \xrightarrow[\varphi[n]]{p} (\mathbb{C}', \sigma')\}.
\end{aligned}
$$

Therefore $\mathit{Level}(\mathbb{C}, \mu, \varphi, n) \overset{\varphi[n]}{\rightsquigarrow} \mathit{Level}(\mathbb{C}, \mu, \varphi, n+1)$.

Lemma 60. *For all* $\mathbb{C}, \mu, \varphi, \vec{W}$*, if* $\mathbf{Nosplit}(\mathbb{C})$ *and* $\boldsymbol{History}(\mathit{init}(\mathbb{C}, \mu), \varphi, \vec{W})$*, then for all* n*,* $\vec{W}[n] = \mathit{Level}(\mathbb{C}, \mu, \varphi, n)$*.*

Proof. For all $\mathbb{C}, \mu, \varphi, \vec{W}$ such that $\mathbf{Nosplit}(\mathbb{C})$ and $\mathbf{History}(\mathit{init}(\mathbb{C}, \mu), \varphi, \vec{W})$, there exists $t, \varphi', \vec{W}'$ such that $\varphi = t :: \varphi'$, $\vec{W} = \mathit{init}(\mathbb{C}, \mu) :: \vec{W}'$, $\mathit{init}(\mathbb{C}, \mu) \overset{t}{\hookrightarrow} W'$ and $\mathbf{History}(W', \varphi', \vec{W}')$. we prove $\vec{W}[n] = \mathit{Level}(\mathbb{C}, \mu, \varphi, n)$ for all n by induction on n.

- base case: $n = 0$.
  $\vec{W}[0] = \mathit{init}(\mathbb{C}, \mu) = \delta(\mathbb{C}) \otimes \mu$, and

$$\begin{aligned}
&\mathit{Level}(\mathbb{C}, \mu, \varphi, 0)\\
&= \lambda(\mathbb{C}', \sigma').\ \textstyle\sum_\sigma \{\mu(\sigma) \cdot p \mid (\mathbb{C}, \sigma) \xrightarrow{p}{}^{0} (\mathbb{C}', \sigma')\}\\
&= \lambda(\mathbb{C}', \sigma').\ \textstyle\sum_\sigma \{\mu(\sigma) \cdot p \mid \mathbb{C} = \mathbb{C}' \wedge \sigma = \sigma' \wedge p = 1\}\\
&= \lambda(\mathbb{C}', \sigma').\ \delta(\mathbb{C})(\mathbb{C}') \cdot \mu(\sigma')\\
&= \delta(\mathbb{C}) \otimes \mu.
\end{aligned}$$

  so $\vec{W}[0] = \mathit{Level}(\mathbb{C}, \mu, \varphi, 0)$.
- inductive case: $n = k + 1$.
  IH: $\vec{W}[k] = \mathit{Level}(\mathbb{C}, \mu, \varphi, k)$.
  From $\mathbf{Nosplit}(\mathbb{C})$ by lem. 56 we know $\mathbf{Nosplit}(\mathit{init}(\mathbb{C}, \mu))$. From $\mathbf{History}(\mathit{init}(\mathbb{C}, \mu), \varphi, \vec{W})$ by Lem. 30 we know $\vec{W}[k] \overset{\varphi[k]}{\rightsquigarrow} \vec{W}[k{+}1]$. By Lem. 59 we know $\mathit{Level}(\mathbb{C}, \mu, \varphi, k) \overset{\varphi[k]}{\rightsquigarrow}$ $\mathit{Level}(\mathbb{C}, \mu, \varphi, k{+}1)$, by IH we have $\vec{W}[k] \overset{\varphi[k]}{\rightsquigarrow} \mathit{Level}(\mathbb{C}, \mu, \varphi, k{+}1)$. By Lem. 57 we know $\vec{W}[k+1] = \mathit{Level}(\mathbb{C}, \mu, \varphi, k+1)$.

Lemma 61. *For all* W *and* t*, there exists* W' *such that* $W \overset{t}{\rightsquigarrow} W'$*.*

Proof. Let $W' \overset{\text{def}}{=} \lambda(\mathbb{C}', \sigma').\ \sum_{\mathbb{C},\sigma} \{p \cdot W(\mathbb{C}, \sigma) \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')\}$, by definition of $\rightsquigarrow$ we know $W \overset{t}{\rightsquigarrow} W'$.

Lemma 62. *For all* W *and* b*,* $[\![\boldsymbol{Pr}(b)]\!]_{W^{(State)}} = \boldsymbol{Pr}_{(\mathbb{C},\sigma)\sim W}[\sigma \models b]$*.*

Proof. For all W and b, by Lem. 3 we know $[\![\mathbf{Pr}(b)]\!]_{W^{(State)}} = \mathbf{Pr}_{\sigma \sim W^{(State)}}[\sigma \models b] =$
$\mathbf{Pr}_{(\mathbb{C},\sigma)\sim W}[\sigma \models b]$.

Lemma 63. *For all* W *and* b*,* $W|_b$ *exists if and only if* $[\![\boldsymbol{Pr}(b)]\!]_{W^{(State)}} > 0$*.*

Proof. For all W and b, by definition of $W|_b$ we know $W|_b$ exists if and only if $W|_{\lambda(\mathbb{C},\sigma).\ \sigma \models b}$ exists. By Eqn. 2 we know $W|_{\lambda(\mathbb{C},\sigma).\ \sigma \models b}$ exists if and only if $\mathbf{Pr}_{(\mathbb{C},\sigma)\sim W}[\sigma \models b] > 0$. By Lem. 62 we know $\mathbf{Pr}_{(\mathbb{C},\sigma)\sim W}[\sigma \models b] > 0$ if and only if $[\![\mathbf{Pr}(b)]\!]_{W^{(State)}} > 0$. Therefore, $W|_b$ exists if and only if $[\![\mathbf{Pr}(b)]\!]_{W^{(State)}} > 0$.

Lemma 64. *For all W and $b_1, \ldots, b_k$, if* $\mathbf{validsplit}(\boldsymbol{split}(b_1, \ldots, b_k))$, *then* $\sum_{i=1}^{k} [\![\boldsymbol{Pr}(b_i)]\!]_{W^{(State)}} = 1$.

Proof. For all W and $b_1, \ldots, b_k$ such that $\mathbf{validsplit}(\mathbf{split}(b_1, \ldots, b_k))$, we know for all σ, $\forall i, j.\ i \neq j \implies \neg(\sigma \models b_i \wedge \sigma \models b_j)$ and $\sigma \models b_1 \vee \ldots b_k$, thus

$$
\begin{aligned}
& \textstyle\sum_{i=1}^{k} [\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}} \\
= & \textstyle\sum_{i=1}^{k} \mathbf{Pr}_{\sigma \sim W^{(State)}}[\sigma \models b_i] \\
= & \ \mathbf{Pr}_{\sigma \sim W^{(State)}}[(\sigma \models b_1) \vee \cdots \vee (\sigma \models b_k)] \qquad \text{(by Lem. 1)} \\
= & \ \mathbf{Pr}_{\sigma \sim W^{(State)}}[\sigma \models b_1 \vee \cdots \vee b_k] \\
= & \textstyle\sum_{\sigma} \{W^{(State)}(\sigma) \mid \sigma \models b_1 \vee \cdots \vee b_k\} \\
= & \textstyle\sum_{\sigma} W^{(State)}(\sigma) \\
= & \ |W^{(State)}| \\
= & \ 1.
\end{aligned}
$$

Lemma 65. *For all W and $b_1, \ldots, b_k$, if* $\mathbf{validsplit}(\boldsymbol{split}(b_1, \ldots, b_k))$, *then there exists i such that $W|_{b_i}$ exists.*

Proof. For all W and $b_1, \ldots, b_k$ such that $\mathbf{validsplit}(\mathbf{split}(b_1, \ldots, b_k))$, by Lem. 63, we need to prove there exists i such that $[\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}} > 0$. We prove it by contradiction. Assume there is no i such that $[\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}} > 0$, i.e., for all i, $[\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}} = 0$, then $\sum_{i=1}^{k} [\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}} = 0$. From $\mathbf{validsplit}(\mathbf{split}(b_1, \ldots, b_k))$ by Lem. 64 we know $\sum_{i=1}^{k} [\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}} = 1$, which contradicts with $\sum_{i=1}^{k} [\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}} = 1$. Therefore, there exists i such that $[\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}} > 0$.

Lemma 66. *For all W and t, there exists W' such that $W \overset{t}{\hookrightarrow} W'$.*

Proof. For all W and t, by Lem. 61 we know there exists W'' such that $W \overset{t}{\rightsquigarrow} W''$. It is obvious that $\#nextsplit(W) = 1 \vee \#nextsplit(W) > 1$. We prove the two cases respectively.

- case 1: $\#nextsplit(W) = 1$.
  There exists $b_1, \ldots, b_k$ such that $nextsplit(W) = \{\mathbf{split}(b_1, \ldots, b_k)\}$ and $\mathbf{validsplit}(\mathbf{split}(b_1, \ldots, b_k))$.
  By Lem. 65 we know there exists i such that $W''|_{b_i}$ exists. Let $W' \overset{\text{def}}{=} W''|_{b_i}$, from $W \overset{t}{\rightsquigarrow} W''$, $nextsplit(W) = \{\mathbf{split}(b_1, \ldots, b_k)\}$, $\mathbf{validsplit}(\mathbf{split}(b_1, \ldots, b_k))$ and $W''|_{b_i} = W'$ we know $W \overset{t}{\hookrightarrow} W'$.
- case 2: $\#nextsplit(W) > 1$.
  From $W \overset{t}{\rightsquigarrow} W''$ and $\#nextsplit(W) > 1$ we have $W \overset{t}{\hookrightarrow} W''$. Let $W' \overset{\text{def}}{=} W''$, then $W \overset{t}{\hookrightarrow} W'$.

Lemma 67. *For all W_0 and φ, there exists $\overrightarrow{W}$ such that* $\boldsymbol{History}(W_0, \varphi, \overrightarrow{W})$.

Proof. by coinduction. From the definition of *Schedule*, there exists t and φ' such that $\varphi = t :: \varphi'$. By Lem. 66, there exists W_1 such that $W_0 \overset{t}{\hookrightarrow} W_1$. By coinduction hypothesis there eixsts $\vec{W}_1$ such that $\mathbf{History}(W_1|_{b_i}, \varphi', \vec{W}_1)$. From $W_0 \overset{t}{\hookrightarrow} W_1|_{b_i}$ we have $\mathbf{History}(W_0, t :: \varphi', W_0 :: \vec{W}_1)$. Let $\vec{W} \overset{\text{def}}{=} W_0 :: \vec{W}_1$, then $\mathbf{History}(W_0, \varphi, \vec{W})$.

Lemma 68. *For all $P, \mathbb{C}, Q$, if $\models_A \{P\}\mathbb{C}\{Q\}$ and $\mathbf{Nosplit}(\mathbb{C})$, then $\models \{P\}\mathbb{C}\{Q\}$.*

Proof. For all $P, \mathbb{C}, Q$ such that $\models_A \{P\}\mathbb{C}\{Q\}$ and $\mathbf{Nosplit}(\mathbb{C})$, by Lem. 44 we know $\models_{A'} \{P\}\mathbb{C}\{Q\}$. We need to prove for all μ and φ such that $\mu \models P$ and $|[\![\mathbb{C}]\!]_\varphi(\mu)| = 1$, then $[\![\mathbb{C}]\!]_\varphi(\mu) \models Q$. By Lem. 67 we know there exists $\vec{W}$ such that $\mathbf{History}(\mathit{init}(\mathbb{C}, \mu), \varphi, \vec{W})$. By Lem. 60 we have $\forall n.\ \vec{W}[n] = \mathit{Level}(\mathbb{C}, \mu, \varphi, n)$. Therefore $\forall n.\ \vec{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = \mathit{Level}(\mathbb{C}, \mu, \varphi, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})$, so

$$
\begin{aligned}
&\lim_{n\to\infty} \vec{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \\
&= \lim_{n\to\infty} \mathit{Level}(\mathbb{C}, \mu, \varphi, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \\
&= \lim_{n\to\infty} \textstyle\sum_{\sigma'} \mathit{Level}(\mathbb{C}, \mu, \varphi, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma') \\
&= \lim_{n\to\infty} \textstyle\sum_{\sigma'} \mathbb{E}_{\sigma\sim\mu}[p \mid (\mathbb{C}, \sigma) \xrightarrow[\varphi]{p}{}^{n} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma')] \\
&= \textstyle\sum_{\sigma'} \lim_{n\to\infty} \mathbb{E}_{\sigma\sim\mu}[p \mid (\mathbb{C}, \sigma) \xrightarrow[\varphi]{p}{}^{n} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma')] \\
&= \textstyle\sum_{\sigma'} \mathbb{E}_{\sigma\sim\mu}[\lim_{n\to\infty} p \mid (\mathbb{C}, \sigma) \xrightarrow[\varphi]{p}{}^{n} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma')] \\
&= \textstyle\sum_{\sigma'} \mathbb{E}_{\sigma\sim\mu}[[\![\mathbb{C}]\!]_\varphi(\sigma)(\sigma')] \\
&= \textstyle\sum_{\sigma'} \mathbb{E}_{\sigma\sim\mu}\{[\![\mathbb{C}]\!]_\varphi(\sigma)\}(\sigma') \\
&= \textstyle\sum_{\sigma'} [\![\mathbb{C}]\!]_\varphi(\mu)(\sigma') \\
&= |[\![\mathbb{C}]\!]_\varphi(\mu)| \\
&= 1.
\end{aligned}
$$

From $\forall n.\ \vec{W}[n] = \mathit{Level}(\mathbb{C}, \mu, \varphi, n)$ we know $\forall \sigma', n.\ \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma') = \mathit{Level}(\mathbb{C}, \mu, \varphi, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma')$, so

$$
\begin{aligned}
&\lim_{n\to\infty} \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma') \\
&= \lim_{n\to\infty} \mathit{Level}(\mathbb{C}, \mu, \varphi, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma') \\
&= \lim_{n\to\infty} \mathbb{E}_{\sigma\sim\mu}[p \mid (\mathbb{C}, \sigma) \xrightarrow[\varphi]{p}{}^{n} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma')] \\
&= \mathbb{E}_{\sigma\sim\mu}[\lim_{n\to\infty} p \mid (\mathbb{C}, \sigma) \xrightarrow[\varphi]{p}{}^{n} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma')] \\
&= \mathbb{E}_{\sigma\sim\mu}[[\![\mathbb{C}]\!]_\varphi(\sigma)(\sigma')] \\
&= \mathbb{E}_{\sigma\sim\mu}\{[\![\mathbb{C}]\!]_\varphi(\sigma)\}(\sigma') \\
&= [\![\mathbb{C}]\!]_\varphi(\mu)(\sigma')
\end{aligned}
$$

holds for all σ'.
Therefore $init(\mathbb{C}, \mu) \Downarrow_{\varphi} [\![\mathbb{C}]\!]_{\varphi}(\mu)$. From $\mu \models P$ and $\models_{A'} \{P\}\mathbb{C}\{Q\}$ we know $[\![\mathbb{C}]\!]_{\varphi}(\mu) \models Q$.

Lemma 69. *For all* $C, \sigma, C', \sigma', p$, *if* $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (C', \sigma')$, *then there exists unique* C'' *such that* $(C, \sigma) \xrightarrow{p} (C'', \sigma') \wedge C' = \mathbf{RemoveSplit}(C'')$.

Proof. by induction on the structure of C.

- case 1: $C = \mathbf{skip}$.
  $\mathbf{RemoveSplit}(C) = \mathbf{skip}$. From $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (C', \sigma')$ we know $C' = \mathbf{skip}, \sigma' = \sigma, p = 1$. Let $C'' \stackrel{\text{def}}{=} \mathbf{skip}$, so $\mathbf{RemoveSplit}(C'') = \mathbf{skip} = C'$.
  From $(\mathbf{skip}, \sigma) \xrightarrow{1} (\mathbf{skip}, \sigma)$ we have $(C, \sigma) \xrightarrow{p} (C'', \sigma')$.
- case 2: $C = x := e$.
  $\mathbf{RemoveSplit}(C) = x := e$. From $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (C', \sigma')$ we know
  $C' = \mathbf{skip}, \sigma' = \sigma\{x \rightsquigarrow [\![e]\!]_{\sigma}\}, p = 1$. Let $C'' \stackrel{\text{def}}{=} \mathbf{skip}$, so $\mathbf{RemoveSplit}(C'') = \mathbf{skip} = C'$.
  From $(x := e, \sigma) \xrightarrow{1} (\mathbf{skip}, \sigma\{x \rightsquigarrow [\![e]\!]_{\sigma}\})$ we have $(C, \sigma) \xrightarrow{p} (C'', \sigma')$.
- case 3: $C = C_1; C_2$.
  IH: For all σ, C', σ', p, if $(\mathbf{RemoveSplit}(C_1), \sigma) \xrightarrow{p} (C', \sigma')$, then there exists unique C'' such that $(C_1, \sigma) \xrightarrow{p} (C'', \sigma') \wedge C' = \mathbf{RemoveSplit}(C'')$.
  It is obvious that $C_1 = \mathbf{skip} \vee C_1 \neq \mathbf{skip}$, we prove the two cases respectively.
  * case 3.1: $C_1 = \mathbf{skip}$.
    $\mathbf{RemoveSplit}(C) = \mathbf{RemoveSplit}(\mathbf{skip}; C_2) = \mathbf{skip}; \mathbf{RemoveSplit}(C_2)$.
    From $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (C', \sigma')$ we know $C' = \mathbf{RemoveSplit}(C_2), \sigma' = \sigma, p = 1$.
    Let $C'' \stackrel{\text{def}}{=} C_2$, so $\mathbf{RemoveSplit}(C'') = \mathbf{RemoveSplit}(C_2) = C'$.
    From $(\mathbf{skip}; C_2, \sigma) \xrightarrow{1} (C_2, \sigma)$ we have $(C, \sigma) \xrightarrow{p} (C'', \sigma')$.
  * case 3.2: $C_1 \neq \mathbf{skip}$.
    $\mathbf{RemoveSplit}(C) = \mathbf{RemoveSplit}(C_1; C_2) = \mathbf{RemoveSplit}(C_1); \mathbf{RemoveSplit}(C_2)$.
    From $C_1 \neq \mathbf{skip}$ we know $\mathbf{RemoveSplit}(C_1) \neq \mathbf{skip}$. From $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (C', \sigma')$ we know there exists unique C_1' such that $C' = C_1'; \mathbf{RemoveSplit}(C_2)$ and
    $(\mathbf{RemoveSplit}(C_1), \sigma) \xrightarrow{p} (C_1', \sigma')$, by IH there exists unique C_1'' such that
    $(C_1, \sigma) \xrightarrow{p} (C_1'', \sigma') \wedge C' = \mathbf{RemoveSplit}(C_1'')$.
    Let $C'' \stackrel{\text{def}}{=} C_1''; C_2$, so $\mathbf{RemoveSplit}(C'') = \mathbf{RemoveSplit}(C_1''); \mathbf{RemoveSplit}(C_2) =$
    $C_1'; \mathbf{RemoveSplit}(C_2) = C'$.
    From $(C_1, \sigma) \xrightarrow{p} (C_1'', \sigma')$ and $C_1 \neq \mathbf{skip}$ we have $(C_1; C_2, \sigma) \xrightarrow{p} (C_1''; C_2, \sigma')$, so $(C, \sigma) \xrightarrow{p} (C'', \sigma')$.

- case 4: $C = \textbf{if}\ (b)\ \textbf{then}\ C_1\ \textbf{else}\ C_2$.
  $\textbf{RemoveSplit}(C) = \textbf{if}\ (b)\ \textbf{then}\ \textbf{RemoveSplit}(C_1)\ \textbf{else}\ \textbf{RemoveSplit}(C_2)$.
  It is obvious that $[\![b]\!]_\sigma = \mathrm{tt} \vee [\![b]\!]_\sigma = \mathrm{ff}$, we prove the two cases respectively.
  * case 4.1: $[\![b]\!]_\sigma = \mathrm{tt}$.
    From $(\textbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (C', \sigma')$ we know $C' = \textbf{RemoveSplit}(C_1), \sigma' = \sigma, p = 1$. Let $C'' \stackrel{\text{def}}{=} C_1$, so $\textbf{RemoveSplit}(C'') = \textbf{RemoveSplit}(C_1) = C'$. From $[\![b]\!]_\sigma = \mathrm{tt}$ we know $(\textbf{if}\ (b)\ \textbf{then}\ C_1\ \textbf{else}\ C_2, \sigma) \xrightarrow{1} (C_1, \sigma)$, so $(C, \sigma) \xrightarrow{p} (C'', \sigma')$.
  * case 4.2: $[\![b]\!]_\sigma = \mathrm{ff}$.
    From $(\textbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (C', \sigma')$ we know $C' = \textbf{RemoveSplit}(C_2), \sigma' = \sigma, p = 1$. Let $C'' \stackrel{\text{def}}{=} C_2$, so $\textbf{RemoveSplit}(C'') = \textbf{RemoveSplit}(C_2) = C'$. From $[\![b]\!]_\sigma = \mathrm{ff}$ we know $(\textbf{if}\ (b)\ \textbf{then}\ C_1\ \textbf{else}\ C_2, \sigma) \xrightarrow{1} (C_2, \sigma)$, so $(C, \sigma) \xrightarrow{p} (C'', \sigma')$.
- case 5: $C = \textbf{while}\ (b)\ \textbf{do}\ C_1$.
  $\textbf{RemoveSplit}(C) = \textbf{while}\ (b)\ \textbf{do}\ \textbf{RemoveSplit}(C_1)$.
  It is obvious that $[\![b]\!]_\sigma = \mathrm{tt} \vee [\![b]\!]_\sigma = \mathrm{ff}$, we prove the two cases respectively.
  * case 5.1: $[\![b]\!]_\sigma = \mathrm{tt}$.
    From $(\textbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (C', \sigma')$ we know
    $C' = \textbf{RemoveSplit}(C_1); \textbf{while}\ (b)\ \textbf{do}\ \textbf{RemoveSplit}(C_1), \sigma' = \sigma, p = 1$.
    Let $C'' \stackrel{\text{def}}{=} C_1; \textbf{while}\ (b)\ \textbf{do}\ C_1$, so $\textbf{RemoveSplit}(C'') =$
    $\textbf{RemoveSplit}(C_1); \textbf{while}\ (b)\ \textbf{do}\ \textbf{RemoveSplit}(C_1) = C'$.
    From $[\![b]\!]_\sigma = \mathrm{tt}$ we know $(\textbf{while}\ (b)\ \textbf{do}\ C_1, \sigma) \xrightarrow{1} (C_1; \textbf{while}\ (b)\ \textbf{do}\ C_1, \sigma)$, so $(C, \sigma) \xrightarrow{p} (C'', \sigma')$.
  * case 5.2: $[\![b]\!]_\sigma = \mathrm{ff}$.
    From $(\textbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (C', \sigma')$ we know $C' = \textbf{skip}, \sigma' = \sigma, p = 1$.
    Let $C'' \stackrel{\text{def}}{=} \textbf{skip}$, so $\textbf{RemoveSplit}(C'') = \textbf{skip}$.
    From $[\![b]\!]_\sigma = \mathrm{ff}$ we know $(\textbf{while}\ (b)\ \textbf{do}\ C_1, \sigma) \xrightarrow{1} (\textbf{skip}, \sigma)$, so $(C, \sigma) \xrightarrow{p} (C'', \sigma')$.
- case 6: $C = \langle C_1 \rangle$.
  $\textbf{RemoveSplit}(C) = \langle C_1 \rangle = C$. From $(\textbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (C', \sigma')$ we know $C' = \textbf{skip}$. Let $C'' \stackrel{\text{def}}{=} \textbf{skip}$, then $\textbf{RemoveSplit}(C'') = \textbf{skip} = C'$ and $(C, \sigma) \xrightarrow{p} (C'', \sigma')$.
- case 7: $C = \langle C_1 \rangle\ sp$.
  $\textbf{RemoveSplit}(C) = \langle C_1 \rangle$. From $(\textbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (C', \sigma')$ we know $C' = \textbf{skip}$,
  so $(\langle C_1 \rangle, \sigma) \xrightarrow{p} (\textbf{skip}, \sigma')$. Let $C'' \stackrel{\text{def}}{=} \textbf{skip}$, then $\textbf{RemoveSplit}(C'') = \textbf{skip} = C'$.
  From $(\langle C_1 \rangle, \sigma) \xrightarrow{p} (\textbf{skip}, \sigma')$ we know $(\langle C_1 \rangle\ sp, \sigma) \xrightarrow{p} (\textbf{skip}, \sigma')$, so $(C, \sigma) \xrightarrow{p} (C'', \sigma')$.
- case 8: $C = \langle C_1 \rangle \oplus_{p'} \langle C_2 \rangle$.
  $\textbf{RemoveSplit}(C) = \langle C_1 \rangle \oplus_{p'} \langle C_2 \rangle = C$. From $(\textbf{RemoveSplit}(C), \sigma) \xrightarrow{p}$

(C', σ') we know
$C' = \langle C_1 \rangle \wedge p = p' \wedge \sigma' = \sigma$ or $C' = \langle C_2 \rangle \wedge p = 1 - p' \wedge \sigma' = \sigma$.
We prove the two cases respectively.
* case 8.1: $C' = \langle C_1 \rangle \wedge p = p' \wedge \sigma' = \sigma$.
  Let $C'' = \langle C_1 \rangle$, then $\mathbf{RemoveSplit}(C'') = \langle C_1 \rangle = C'$. From $(\langle C_1 \rangle \oplus_{p'} \langle C_2 \rangle, \sigma) \xrightarrow{p'} (\langle C_1 \rangle, \sigma)$ we know $(C, \sigma) \xrightarrow{p} (C'', \sigma')$.
* case 8.1: $C' = \langle C_2 \rangle \wedge p = 1 - p' \wedge \sigma' = \sigma$.
  Let $C'' = \langle C_2 \rangle$, then $\mathbf{RemoveSplit}(C'') = \langle C_2 \rangle = C'$. From $(\langle C_1 \rangle \oplus_{p'} \langle C_2 \rangle, \sigma) \xrightarrow{1-p'} (\langle C_2 \rangle, \sigma)$ we know $(C, \sigma) \xrightarrow{p} (C'', \sigma')$.

Lemma 70. *For all $C, \sigma, C', \sigma', p$, if $(C, \sigma) \xrightarrow{p} (C', \sigma')$, then $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (\mathbf{RemoveSplit}(C'), \sigma')$.*

Proof. by induction on the derivation of $(C, \sigma) \xrightarrow{p} (C', \sigma')$.

- case 1: $C = C' = \mathbf{skip}, \sigma = \sigma', p = 1$.
  $\mathbf{RemoveSplit}(C) = \mathbf{RemoveSplit}(C') = \mathbf{skip}$.
  From $(\mathbf{skip}, \sigma) \xrightarrow{1} (\mathbf{skip}, \sigma)$ we know $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (\mathbf{RemoveSplit}(C'), \sigma')$.
- case 2: $C = x := e, C' = \mathbf{skip}, \sigma' = \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}, p = 1$.
  $\mathbf{RemoveSplit}(C) = x := e$. $\mathbf{RemoveSplit}(C') = \mathbf{skip}$.
  From $(x := e, \sigma) \xrightarrow{1} (\mathbf{skip}, \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})$ we know
  $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (\mathbf{RemoveSplit}(C'), \sigma')$.
- case 3: $C = \mathbf{skip}; C_2, C' = C_2, \sigma = \sigma', p = 1$.
  $\mathbf{RemoveSplit}(C) = \mathbf{skip}; \mathbf{RemoveSplit}(C_2)$. $\mathbf{RemoveSplit}(C') = \mathbf{RemoveSplit}(C_2)$.
  From $(\mathbf{skip}; \mathbf{RemoveSplit}(C_2), \sigma) \xrightarrow{1} (\mathbf{RemoveSplit}(C_2), \sigma)$ we know
  $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (\mathbf{RemoveSplit}(C'), \sigma')$.
- case 4: $C = C_1; C_2, C_1 \neq \mathbf{skip}, C' = C_1'; C_2, (C_1, \sigma) \xrightarrow{p} (C_1', \sigma')$.
  IH: $(\mathbf{RemoveSplit}(C_1), \sigma) \xrightarrow{p} (\mathbf{RemoveSplit}(C_1'), \sigma')$.
  $\mathbf{RemoveSplit}(C) = \mathbf{RemoveSplit}(C_1); \mathbf{RemoveSplit}(C_2)$.
  $\mathbf{RemoveSplit}(C') = \mathbf{RemoveSplit}(C_1'); \mathbf{RemoveSplit}(C_2)$.
  From $C_1 \neq \mathbf{skip}$ we know $\mathbf{RemoveSplit}(C_1) \neq \mathbf{skip}$.
  From $(\mathbf{RemoveSplit}(C_1), \sigma) \xrightarrow{p} (\mathbf{RemoveSplit}(C_1'), \sigma')$ we know
  $(\mathbf{RemoveSplit}(C_1); \mathbf{RemoveSplit}(C_2), \sigma) \xrightarrow{p} (\mathbf{RemoveSplit}(C_1'); \mathbf{RemoveSplit}(C_2), \sigma)$,
  so
  $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (\mathbf{RemoveSplit}(C'), \sigma')$.
- case 5: $C = \mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2, [\![b]\!]_\sigma = \mathrm{tt}, C' = C_1, \sigma' = \sigma, p = 1$.
  $\mathbf{RemoveSplit}(C) = \mathbf{if}\ (b)\ \mathbf{then}\ \mathbf{RemoveSplit}(C_1)\ \mathbf{else}\ \mathbf{RemoveSplit}(C_2)$.
  $\mathbf{RemoveSplit}(C') = \mathbf{RemoveSplit}(C_1)$.
  From $[\![b]\!]_\sigma = \mathrm{tt}$ we know
  $(\mathbf{if}\ (b)\ \mathbf{then}\ \mathbf{RemoveSplit}(C_1)\ \mathbf{else}\ \mathbf{RemoveSplit}(C_2), \sigma) \xrightarrow{1} (\mathbf{RemoveSplit}(C_1), \sigma)$,
  so
  $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (\mathbf{RemoveSplit}(C'), \sigma')$.
- case 6: $C = \mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2, [\![b]\!]_\sigma = \mathrm{ff}, C' = C_2, \sigma' = \sigma, p = 1$.
  $\mathbf{RemoveSplit}(C) = \mathbf{if}\ (b)\ \mathbf{then}\ \mathbf{RemoveSplit}(C_1)\ \mathbf{else}\ \mathbf{RemoveSplit}(C_2)$.

$\mathbf{RemoveSplit}(C') = \mathbf{RemoveSplit}(C_2)$.
From $[\![b]\!]_\sigma = \mathrm{ff}$ we know
$(\mathbf{if}\ (b)\ \mathbf{then}\ \mathbf{RemoveSplit}(C_1)\ \mathbf{else}\ \mathbf{RemoveSplit}(C_2), \sigma) \xrightarrow{1} (\mathbf{RemoveSplit}(C_2), \sigma)$,
so
$(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (\mathbf{RemoveSplit}(C'), \sigma')$.

- case 7: $C = \mathbf{while}\ (b)\ \mathbf{do}\ C_1, [\![b]\!]_\sigma = \mathrm{tt}, C' = C_1; \mathbf{while}\ (b)\ \mathbf{do}\ C_1, \sigma' = \sigma, p = 1$.
$\mathbf{RemoveSplit}(C) = \mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{RemoveSplit}(C_1)$.
$\mathbf{RemoveSplit}(C') = \mathbf{RemoveSplit}(C_1); \mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{RemoveSplit}(C_1)$.
From $[\![b]\!]_\sigma = \mathrm{tt}$ we know
$(\mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{RemoveSplit}(C_1), \sigma) \xrightarrow{1} (\mathbf{RemoveSplit}(C_1); \mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{RemoveSplit}(C_1), \sigma)$,
so $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (\mathbf{RemoveSplit}(C'), \sigma')$.
- case 8: $C = \mathbf{while}\ (b)\ \mathbf{do}\ C_1, [\![b]\!]_\sigma = \mathrm{ff}, C' = \mathbf{skip}, \sigma' = \sigma, p = 1$.
$\mathbf{RemoveSplit}(C) = \mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{RemoveSplit}(C_1)$.
$\mathbf{RemoveSplit}(C') = \mathbf{skip};$.
From $[\![b]\!]_\sigma = \mathrm{ff}$ we know $(\mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{RemoveSplit}(C_1), \sigma) \xrightarrow{1} (\mathbf{skip}, \sigma)$,
so $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (\mathbf{RemoveSplit}(C'), \sigma')$.
- case 9: $C = \langle C_1 \rangle, C' = \mathbf{skip}$.
$\mathbf{RemoveSplit}(C) = \langle C_1 \rangle = C$. $\mathbf{RemoveSplit}(C') = \mathbf{skip} = C'$.
From $(C, \sigma) \xrightarrow{p} (C', \sigma')$ we know $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (\mathbf{RemoveSplit}(C'), \sigma')$.
- case 10: $C = \langle C_1 \rangle\ sp, C' = \mathbf{skip}, (\langle C_1 \rangle, \sigma) \xrightarrow{p} (\mathbf{skip}, \sigma')$.
$\mathbf{RemoveSplit}(C) = \langle C_1 \rangle$. $\mathbf{RemoveSplit}(C') = \mathbf{skip}$.
From $(\langle C_1 \rangle, \sigma) \xrightarrow{p} (\mathbf{skip}, \sigma')$ we know $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p} (\mathbf{RemoveSplit}(C'), \sigma')$.
- case 11: $C = \langle C_1 \rangle \oplus_{p'} \langle C_2 \rangle, C' = \langle C_1 \rangle, \sigma = \sigma', p = p'$.
$\mathbf{RemoveSplit}(C) = \langle C_1 \rangle \oplus_{p'} \langle C_2 \rangle$. $\mathbf{RemoveSplit}(C') = \langle C_1 \rangle$.
From $(\langle C_1 \rangle \oplus_{p'} \langle C_2 \rangle, \sigma) \xrightarrow{p'} (\langle C_1 \rangle, \sigma)$ we know $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p}$ $(\mathbf{RemoveSplit}(C'), \sigma')$.
- case 12: $C = \langle C_1 \rangle \oplus_{p'} \langle C_2 \rangle, C' = \langle C_2 \rangle, \sigma = \sigma', p = 1 - p'$.
$\mathbf{RemoveSplit}(C) = \langle C_1 \rangle \oplus_{p'} \langle C_2 \rangle$. $\mathbf{RemoveSplit}(C') = \langle C_2 \rangle$.
From $(\langle C_1 \rangle \oplus_{p'} \langle C_2 \rangle, \sigma) \xrightarrow{1-p'} (\langle C_2 \rangle, \sigma)$ we know $(\mathbf{RemoveSplit}(C), \sigma) \xrightarrow{p}$ $(\mathbf{RemoveSplit}(C'), \sigma')$.

Lemma 71. *For all* $t, \mathbb{C}, \sigma, \mathbb{C}', \sigma', p$, $(\mathbf{RemoveSplit}(\mathbb{C}), \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')$ *iff there exists unique* $\mathbb{C}''$ *such that* $(\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}'', \sigma') \wedge \mathbb{C}' = \mathbf{RemoveSplit}(\mathbb{C}'')$.

Proof. First we prove if $(\mathbf{RemoveSplit}(\mathbb{C}), \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')$, then there exists unique $\mathbb{C}''$ such that $(\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}'', \sigma') \wedge \mathbb{C}' = \mathbf{RemoveSplit}(\mathbb{C}'')$. By definition of *Prog*, there exists unique $C_1, \ldots, C_n$ such that $\mathbb{C} = C_1 \parallel \cdots \parallel C_n$, so $\mathbf{RemoveSplit}(\mathbb{C}) = \mathbf{RemoveSplit}(C_1) \parallel \cdots \parallel \mathbf{RemoveSplit}(C_n)$. From $\mathbf{RemoveSplit}(C_1) \parallel \cdots \parallel \mathbf{RemoveSplit}(C_n) \xrightarrow[t]{p} (C', \sigma')$ we know there exists unique C'_t such that $\mathbb{C}' = \mathbf{RemoveSplit}(C_1) \parallel \cdots \parallel \mathbf{RemoveSplit}(C_{t-1}) \parallel C'_t \parallel \mathbf{RemoveSplit}(C_{t+1}) \parallel \cdots \parallel \mathbf{RemoveSplit}(C_n)$ and $(\mathbf{RemoveSplit}(C_t), \sigma) \xrightarrow{p}$

(C'_t, σ'). By Lem. 69 we know there exists unique C''_t such that $(C_t, \sigma) \xrightarrow{p} (C''_t, \sigma') \wedge C'_t = \mathbf{RemoveSplit}(C''_t)$. Let $\mathbb{C}'' \stackrel{\text{def}}{=} C_1 \parallel \cdots \parallel C_{t-1} \parallel C''_t \parallel C_{t+1} \parallel \cdots \parallel C_n$. From $(C_t, \sigma) \xrightarrow{p} (C''_t, \sigma')$ we know $(C_1 \parallel \cdots \parallel C_n, \sigma) \xrightarrow[t]{p} (C_1 \parallel \cdots \parallel C_{t-1} \parallel C''_t \parallel C_{t+1} \parallel \cdots \parallel C_n, \sigma')$, i.e., $(\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}'', \sigma')$. From $C'_t = \mathbf{RemoveSplit}(C''_t)$ we know

$$\begin{aligned}
&\mathbf{RemoveSplit}(\mathbb{C}'') \\
&= \mathbf{RemoveSplit}(C_1 \parallel \cdots \parallel C_{t-1} \parallel C'_t \parallel C_{t+1} \parallel \cdots \parallel C_n) \\
&= \mathbf{RemoveSplit}(C_1) \parallel \cdots \parallel \mathbf{RemoveSplit}(C_{t-1}) \parallel \mathbf{RemoveSplit}(C''_t) \parallel \\
&\quad \mathbf{RemoveSplit}(C_{t+1}) \parallel \cdots \parallel \mathbf{RemoveSplit}(C_n) \\
&= \mathbf{RemoveSplit}(C_1) \parallel \cdots \parallel \mathbf{RemoveSplit}(C_{t-1}) \parallel C'_t \parallel \\
&\quad \mathbf{RemoveSplit}(C_{t+1}) \parallel \cdots \parallel \mathbf{RemoveSplit}(C_n) \\
&= \mathbb{C}'.
\end{aligned}$$

Then we prove if there exists unique $\mathbb{C}''$ such that $(\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}'', \sigma') \wedge \mathbb{C}' = \mathbf{RemoveSplit}(\mathbb{C}'')$, then $(\mathbf{RemoveSplit}(\mathbb{C}), \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')$. From $(\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}'', \sigma')$ we know there exists $C_1, \ldots, C_n, C'_t$ such that $\mathbb{C} = C_1 \parallel \cdots \parallel C_n$, $\mathbb{C}'' = C_1 \parallel \cdots \parallel C_{t-1} \parallel C''_t \parallel C_{t+1} \parallel \cdots \parallel C_n$ and $(C_t, \sigma) \xrightarrow[t]{p} (C''_t, \sigma')$. From $\mathbb{C} = C_1 \parallel \cdots \parallel C_n$ we know $\mathbf{RemoveSplit}(\mathbb{C}) = \mathbf{RemoveSplit}(C_1) \parallel \cdots \parallel \mathbf{RemoveSplit}(C_n)$. From $\mathbb{C}'' = C_1 \parallel \cdots \parallel C_{t-1} \parallel C''_t \parallel C_{t+1} \parallel \cdots \parallel C_n$ we know $\mathbf{RemoveSplit}(\mathbb{C}'') = \mathbf{RemoveSplit}(C_1) \parallel \cdots \parallel \mathbf{RemoveSplit}(C_{t-1}) \parallel \mathbf{RemoveSplit}(C''_t) \parallel \mathbf{RemoveSplit}(C_{t+1}) \parallel \cdots \parallel \mathbf{RemoveSplit}(C_n)$. From $(C_t, \sigma) \xrightarrow[t]{p} (C''_t, \sigma')$ by Lem. 70 we have $(\mathbf{RemoveSplit}(C_t), \sigma) \xrightarrow{p} (\mathbf{RemoveSplit}(C''_t), \sigma')$, so $(\mathbf{RemoveSplit}(\mathbb{C}), \sigma) \xrightarrow[t]{p} (\mathbf{RemoveSplit}(\mathbb{C}''), \sigma')$. From $\mathbb{C}' = \mathbf{RemoveSplit}(\mathbb{C}'')$ we know $(\mathbf{RemoveSplit}(\mathbb{C}), \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')$.

Lemma 72. *For all* $n, \varphi, \mathbb{C}, \sigma, \sigma', p$, $(\mathbf{RemoveSplit}(\mathbb{C}), \sigma) \xrightarrow[\varphi]{p}{}^{n} (\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}, \sigma') \iff (\mathbb{C}, \sigma) \xrightarrow[\varphi]{p}{}^{n} (\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}, \sigma')$.

Proof. by induction on n.

- base case: $n = 0$.

$$\begin{aligned}
& (\mathbf{RemoveSplit}(\mathbb{C}), \sigma) \xrightarrow[\varphi]{p}{}^{0} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma') \\
\iff\ & (\mathbf{RemoveSplit}(\mathbb{C}) = \mathbf{skip} \parallel \cdots \parallel \mathbf{skip} \wedge \sigma = \sigma' \wedge p = 1) \vee \\
& ((\mathbf{RemoveSplit}(\mathbb{C}) \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip} \vee \sigma \neq \sigma') \wedge p = 0) \\
\iff\ & (\mathbb{C} = \mathbf{skip} \parallel \cdots \parallel \mathbf{skip} \wedge \sigma = \sigma' \wedge p = 1) \vee \\
& ((\mathbb{C} \neq \mathbf{skip} \parallel \cdots \parallel \mathbf{skip} \vee \sigma \neq \sigma') \wedge p = 0) \\
\iff\ & (\mathbb{C}, \sigma) \xrightarrow[\varphi]{p}{}^{0} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma')
\end{aligned}$$

- inductive case: $n = k + 1$. IH: For all $\varphi, \mathbb{C}, \sigma, \sigma', p$, $(\mathbf{RemoveSplit}(\mathbb{C}), \sigma) \xrightarrow[\varphi]{p}{}^{k} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma') \iff (\mathbb{C}, \sigma) \xrightarrow[\varphi]{p}{}^{k} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma')$.

By definition of *Schedule*, there exists t and φ' that $\varphi = t :: \varphi'$.

$$
\begin{aligned}
& (\mathbf{RemoveSplit}(\mathbb{C}), \sigma) \xrightarrow[\varphi]{p}{}^{n} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma') \\
\iff & (\mathbf{RemoveSplit}(\mathbb{C}), \sigma) \xrightarrow[t::\varphi']{p}{}^{k+1} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma') \\
\iff & p = \textstyle\sum_{\mathbb{C}', \sigma''} \{p_1 \cdot p_2 \mid (\mathbf{RemoveSplit}(\mathbb{C}), \sigma) \xrightarrow[t]{p_1} (\mathbb{C}', \sigma'') \wedge (\mathbb{C}', \sigma'') \xrightarrow[\varphi']{p_2}{}^{k} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma')\} \\
\iff & p = \textstyle\sum_{\mathbb{C}', \sigma''} \{p_1 \cdot p_2 \mid \exists! \mathbb{C}''.\ (\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}'', \sigma'') \wedge \mathbb{C}' = \mathbf{RemoveSplit}(\mathbb{C}'') \wedge \\
& \qquad\qquad (\mathbb{C}', \sigma'') \xrightarrow[\varphi']{p_2}{}^{k} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma')\} \qquad \text{(by Lem. 71)} \\
\iff & p = \textstyle\sum_{\mathbb{C}'', \sigma''} \{p_1 \cdot p_2 \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p_1} (\mathbb{C}'', \sigma'') \wedge (\mathbf{RemoveSplit}(\mathbb{C}''), \sigma'') \xrightarrow[\varphi']{p_2}{}^{k} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma')\} \\
\iff & p = \textstyle\sum_{\mathbb{C}'', \sigma''} \{p_1 \cdot p_2 \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p_1} (\mathbb{C}'', \sigma'') \wedge (\mathbb{C}'', \sigma'') \xrightarrow[\varphi']{p_2}{}^{k} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma')\} \qquad \text{(by IH)} \\
\iff & (\mathbb{C}, \sigma) \xrightarrow[t::\varphi']{p}{}^{k+1} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma') \\
\iff & (\mathbb{C}, \sigma) \xrightarrow[\varphi]{p}{}^{n} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma')
\end{aligned}
$$

Lemma 73. *For all* $\mathbb{C}, \sigma$, $[\![\mathbf{RemoveSplit}(\mathbb{C})]\!]_\varphi(\sigma) = [\![\mathbb{C}]\!]_\varphi(\sigma)$.

Proof. For any $\mathbb{C}, \sigma$, we have

$$
\begin{aligned}
& [\![\mathbf{RemoveSplit}(\mathbb{C})]\!]_\varphi(\sigma) \\
= {} & \lambda\sigma'.\ \lim \vec{p},\ \text{where } \forall n. (\mathbf{RemoveSplit}(\mathbb{C}), \sigma) \xrightarrow[\varphi]{\vec{p}\,[n]}{}^{n} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma') \qquad \text{(by definition)} \\
= {} & \lambda\sigma'.\ \lim \vec{p},\ \text{where } \forall n. (\mathbb{C}, \sigma) \xrightarrow[\varphi]{\vec{p}\,[n]}{}^{n} (\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma') \qquad \text{(by Lem. 72)} \\
= {} & [\![\mathbb{C}]\!]_\varphi(\sigma) \qquad \text{(by definition)}
\end{aligned}
$$

Lemma 74. *For all* $\mathbb{C}, \mu$, $[\![\mathbf{RemoveSplit}(\mathbb{C})]\!]_\varphi(\mu) = [\![\mathbb{C}]\!]_\varphi(\mu)$.

Proof. For any $\mathbb{C}, \mu$, we have

$$
\begin{aligned}
& [\![\mathbf{RemoveSplit}(\mathbb{C})]\!]_\varphi(\mu) \\
= {} & \mathbb{E}_{\sigma\sim\mu}\{[\![\mathbf{RemoveSplit}(\mathbb{C})]\!]_\varphi(\sigma)\} \qquad \text{(by definition)} \\
= {} & \mathbb{E}_{\sigma\sim\mu}\{[\![\mathbb{C}]\!]_\varphi(\sigma)\} \qquad \text{(by Lem. 73)} \\
= {} & [\![\mathbb{C}]\!]_\varphi(\mu) \qquad \text{(by definition)}
\end{aligned}
$$

Lemma 75. *For all* $P, \mathbb{C}, Q$, *if* $\models \{P\}\mathbf{RemoveSplit}(\mathbb{C})\{Q\}$, *then* $\models \{P\}\mathbb{C}\{Q\}$.

Proof. For any $P, \mathbb{C}, Q$ such that $\models \{P\}\mathbf{RemoveSplit}(\mathbb{C})\{Q\}$, we need to prove for all μ and φ, if $\mu \models P$ and $|[\![\mathbb{C}]\!]_\varphi(\mu)| = 1$, then $[\![\mathbb{C}]\!]_\varphi(\mu) \models Q$. By Lem. 74 we know $[\![\mathbf{RemoveSplit}(\mathbb{C})]\!]_\varphi(\mu) = [\![\mathbb{C}]\!]_\varphi(\mu)$, so $|[\![\mathbf{RemoveSplit}(\mathbb{C})]\!]_\varphi(\mu)| = |[\![\mathbb{C}]\!]_\varphi(\mu)| = 1$. From $\mu \models P$ and $\models \{P\}\mathbf{RemoveSplit}(\mathbb{C})\{Q\}$ we have $[\![\mathbf{RemoveSplit}(\mathbb{C})]\!]_\varphi(\mu) \models Q$, so $[\![\mathbb{C}]\!]_\varphi(\mu) \models Q$.

H.3 Proof of Theorem 2

Proof (Proof of Theorem 2). For any $P, \mathbb{C}, Q$ such that $\vdash_{\text{A}} \{P\}\mathbb{C}\{Q\}$, we prove $\models_A \{P\}\mathbb{C}\{Q\}$ by induction on the derivation of $\vdash_{\text{A}} \{P\}\mathbb{C}\{Q\}$.

- case (P-CSQ): $P \Rightarrow P_1$, $\vdash_{\text{A}} \{P_1\}\mathbb{C}\{Q_1\}$ and $Q_1 \Rightarrow Q$.
  From $\vdash_{\text{A}} \{P_1\}\mathbb{C}\{Q_1\}$ by induction hypothesis we have $\models_{\text{A}} \{P_1\}\mathbb{C}\{Q_1\}$. From $P \Rightarrow P_1$ and $Q_1 \Rightarrow Q$ by Lem. 157 we know $\models_{\text{A}} \{P\}\mathbb{C}\{Q\}$.
- case (BIGCONJ): $P = P_1 \wedge \cdots \wedge P_n$, $Q = Q_1 \wedge \cdots \wedge Q_n$, $\vdash_{\text{A}} \{P_1\}\mathbb{C}\{Q_1\}$, ..., $\vdash_{\text{A}} \{P_n\}\mathbb{C}\{Q_n\}$.
  From $\vdash_{\text{A}} \{P_1\}\mathbb{C}\{Q_1\}$, ..., $\vdash_{\text{A}} \{P_n\}\mathbb{C}\{Q_n\}$ by induction hypothesis we have $\models_{\text{A}} \{P_1\}\mathbb{C}\{Q_1\}$, ..., $\models_{\text{A}} \{P_n\}\mathbb{C}\{Q_n\}$. By Lem. 158 we know $\models_{\text{A}} \{P_1 \wedge \cdots \wedge P_n\}\mathbb{C}\{Q_1 \wedge \cdots \wedge Q_n\}$, i.e., $\models_{\text{A}} \{P\}\mathbb{C}\{Q\}$.
- case (BIGDISJ): $P = P_1 \vee \cdots \vee P_n$, $Q = Q_1 \vee \cdots \vee Q_n$, $\vdash_{\text{A}} \{P_1\}\mathbb{C}\{Q_1\}$, ..., $\vdash_{\text{A}} \{P_n\}\mathbb{C}\{Q_n\}$.
  From $\vdash_{\text{A}} \{P_1\}\mathbb{C}\{Q_1\}$, ..., $\vdash_{\text{A}} \{P_n\}\mathbb{C}\{Q_n\}$ by induction hypothesis we have $\models_{\text{A}} \{P_1\}\mathbb{C}\{Q_1\}$, ..., $\models_{\text{A}} \{P_n\}\mathbb{C}\{Q_n\}$. By Lem. 159 we know $\models_{\text{A}} \{P_1 \vee \cdots \vee P_n\}\mathbb{C}\{Q_1 \vee \cdots \vee Q_n\}$, i.e., $\models_{\text{A}} \{P\}\mathbb{C}\{Q\}$.
- case (REMOVESPLIT): $C = \mathbf{RemoveSplit}(C')$, $\vdash_{\text{A}} \{P\}C'\{Q\}$ and $\mathbf{closed}(Q)$.
  From $\vdash_{\text{A}} \{P\}C'\{Q\}$ by induction hypothesis we have $\models_{\text{A}} \{P\}C'\{Q\}$. From $\mathbf{closed}(Q)$ by Lem. 92 we know $\models_{\text{A}} \{P\}\mathbf{RemoveSplit}(C')\{Q\}$, i.e., $\models_{\text{A}} \{P\}\mathbb{C}\{Q\}$.
- case (LAZYCOIN): $\vdash_{\text{A}} \{P\}\mathbf{lazycoin}(\mathbb{C})\{Q\}$.
  From $\vdash_{\text{A}} \{P\}\mathbf{lazycoin}(\mathbb{C})\{Q\}$ by Lem. 156 we have $\models_{\text{A}} \{P\}\mathbf{lazycoin}(\mathbb{C})\{Q\}$.
- case (PAR): $\mathbb{C} = C_1 \parallel \cdots \parallel C_n$, $P \Rightarrow P_1 \wedge \cdots \wedge P_n$, $Q_1 \wedge \cdots \wedge Q_n \Rightarrow Q$, $R_1, G_1, I \vdash_{\text{NST}} \{P_1\}C_1\{Q_1\}$, ..., $R_n, G_n, I \vdash_{\text{NST}} \{P_n\}C_1\{Q_n\}$, $G_j \Rightarrow R_i$ for all $i \neq j$, $\mathbf{lclosed}(I)$, $\mathbf{lclosed}(Q_1)$, ..., $\mathbf{lclosed}(Q_n)$. From $R_1, G_1, I \vdash_{\text{NST}} \{P_1\}C_1\{Q_1\}, \ldots, R_n, G_n, I \vdash_{\text{NST}} \{P_n\}C_1\{Q_n\}$ by Lem. 304 we know $R_1, G_1, I \models_{\text{NST}} \{P_1\}C_1\{Q_1\}$, ..., $R_n, G_n, I \models_{\text{NST}} \{P_n\}C_1\{Q_n\}$. By Lem. 177 we have $\models_{\text{A}} \{P\}\mathbf{lazycoin}(\mathbb{C})\{Q\}$.

The remainder of this section gives the proofs of the lemmas used in the proof of Theorem 2.

Definition 45. Given W, φ, Γ such that $\mathbf{History}_T(1, W, \varphi, \Gamma)$. Let s be an infinite sequence of natural numbers. We write $W \Downarrow^s_\varphi \mu$ if and only if $\lim_{n\to\infty} \mathit{traverse}(\Gamma, s, n)^{(\mathit{Prog})}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$ and $\forall\sigma.\ \lim_{n\to\infty} \mathit{traverse}(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = \mu(\sigma)$.

Definition 46. $\models_{AT} \{P\}\mathbb{C}\{Q\}$ iff for all μ, if $\mu \models P$, then for all φ, s, and μ', if $\mathit{init}(\mathbb{C}, \mu) \Downarrow^s_\varphi \mu'$, then $\mu' \models Q$.

Lemma 76. *For all p, W, φ, Γ, s, if $\boldsymbol{History}_T(p, W, \varphi, \Gamma)$, then $\boldsymbol{History}(W, \varphi, \lambda n.\mathit{traverse}(\Gamma, s, n))$.*

Proof. by coinduction. For all p, W, φ, Γ, s such that $\mathbf{History}_T(p, W, \varphi, \Gamma)$, there are two cases.

- $\varphi = t :: \varphi'$, $\Gamma = \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k)$, $W \overset{t}{\rightsquigarrow} W'$, $\mathit{nextsplit}(W, t) = \{sp\}$, $\mathit{splitter}(W', sp) = \{(W_0, p_0), \ldots, (W_k, p_k)\}$ and $\forall i.\ \mathbf{History}_T(p{\cdot}p_i, W_i, \varphi', \Gamma_i)$.
  It is obvious that there exists m and s' such that $s = m :: s'$ and $m \leq k$. From $W \overset{t}{\rightsquigarrow} W'$ and $\#\mathit{nextsplit}(W, t) > 1$ we know $W \overset{t}{\hookrightarrow} W'$. from $\mathbf{History}_T(p \cdot$

$$
\begin{array}{rcl}
\mathit{splitter}(W, sp) & ::= & \{(W|_{b_i}, [\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}}) \mid 1 \le i \le k \wedge [\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}} > 0\}, \text{where } sp = \mathbf{split}(b_1, \ldots, b_k) \\
\Gamma & ::= & \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k) \quad \text{(coinductive)} \\
s & ::= & n :: s \quad \text{(coinductive)} \\
\mathit{traverse}(\Gamma, s, n) & \stackrel{\text{def}}{=} & \begin{cases} W, & \text{if } \Gamma = \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k) \wedge n = 0 \\ \mathit{traverse}(\Gamma_m, s', n'), & \text{if } \Gamma = \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k) \wedge s = m :: s' \wedge n = n' + 1 \wedge m \le k \\ \mathit{undefined}, & \text{if } \Gamma = \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k) \wedge s = m :: s' \wedge n = n' + 1 \wedge m > k \end{cases} \\
\mathit{zeros} & \stackrel{\text{def}}{=} & 0 :: \mathit{zeros} \\
\mathit{prob}(\Gamma, s, n) & \stackrel{\text{def}}{=} & \begin{cases} p, & \text{if } \Gamma = \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k) \wedge n = 0 \wedge s = \mathit{zeros} \\ 0, & \text{if } \Gamma = \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k) \wedge n = 0 \wedge s \neq \mathit{zeros} \\ \mathit{prob}(\Gamma_m, s', n'), & \text{if } \Gamma = \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k) \wedge s = m :: s' \wedge n = n' + 1 \wedge m \le k \\ \mathit{undefined}, & \text{if } \Gamma = \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k) \wedge s = m :: s' \wedge n = n' + 1 \wedge m > k \end{cases}
\end{array}
$$

$$
\frac{W \overset{t}{\rightsquigarrow} W' \quad \mathit{nextsplit}(W, t) = \{sp\} \quad \mathit{splitter}(W', sp) = \{(W_0, p_0), \ldots, (W_k, p_k)\} \quad \forall i.\ \mathbf{History}_T(p \cdot p_i, W_i, \varphi, \Gamma_i)}{\mathbf{History}_T(p, W, t :: \varphi, \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k))}
$$

$$
\frac{W \overset{t}{\rightsquigarrow} W' \quad \#\mathit{nextsplit}(W, t) > 1 \quad \mathbf{History}_T(p, W', \varphi, \Gamma)}{\mathbf{History}_T(p, W, t :: \varphi, \mathbf{Tree}(p, W, \Gamma))}
$$

Fig. 33: Auxiliary Definitions in Def. 45

$p_m, W_m, \varphi', \Gamma_m)$ by coinduction hypothesis we know $\mathbf{History}(W_m, \varphi', \lambda n.\mathit{traverse}(\Gamma', s, n))$, thus $\mathbf{History}(W, t :: \varphi', W :: \lambda n.\mathit{traverse}(\Gamma_m, s', n))$. From $\varphi = t :: \varphi'$ and

$$
\begin{array}{rl}
& \lambda n.\mathit{traverse}(\Gamma, s, n) \\
= & \lambda n.\mathit{traverse}(\mathbf{Tree}(p, W, \Gamma_1, \ldots, \Gamma_k), m :: s', n) \\
= & \lambda n. \begin{cases} W, & \text{if } n = 0 \\ \mathit{traverse}(\Gamma_m, s', n'), & \text{if } n = n' + 1 \end{cases} \\
= & W :: \lambda n.\mathit{traverse}(\Gamma_m, s', n)
\end{array}
$$

we know $\mathbf{History}(W, \varphi, \lambda n.\mathit{traverse}(\Gamma, s, n))$.

- $\varphi = t :: \varphi'$, $\Gamma = \mathbf{Tree}(p, W, \Gamma')$, $W \overset{t}{\rightsquigarrow} W'$, $\#\mathit{nextsplit}(W, t) > 1$ and $\mathbf{History}_T(p, W', \varphi', \Gamma')$.

  It is obvious that there exists s' such that $s = 0 :: s'$. From $W \overset{t}{\rightsquigarrow} W'$ and $\#\mathit{nextsplit}(W, t) > 1$ we know $W \overset{t}{\hookrightarrow} W'$. From $\mathbf{History}_T(p, W', \varphi', \Gamma')$ by coinduction hypothesis we know $\mathbf{History}(W', \varphi', \lambda n.\mathit{traverse}(\Gamma', s, n))$, thus $\mathbf{History}(W, t :: \varphi', W :: \lambda n.\mathit{traverse}(\Gamma', s', n))$. From $\varphi = t :: \varphi'$ and

$$
\begin{array}{rl}
& \lambda n.\mathit{traverse}(\Gamma, s, n) \\
= & \lambda n.\mathit{traverse}(\mathbf{Tree}(p, W, \Gamma'), 0 :: s', n) \\
= & \lambda n. \begin{cases} W, & \text{if } n = 0 \\ \mathit{traverse}(\Gamma', s', n'), & \text{if } n = n' + 1 \end{cases} \\
= & W :: \lambda n.\mathit{traverse}(\Gamma', s', n)
\end{array}
$$

we know $\mathbf{History}(W, \varphi, \lambda n.traverse(\Gamma, s, n))$.

Lemma 77. *For all W, φ, s, μ, if $W \Downarrow^s_\varphi \mu$, then $W \Downarrow'_\varphi \mu$.*

Proof. For all W, φ, s, μ such that $W \Downarrow^s_\varphi \mu$, by Def. 45 we know there exists Γ such that $\mathbf{History}_T(1, W, \varphi, \Gamma)$, $\lim_{n\to\infty} traverse(\Gamma, s, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$ and $\forall\sigma. \lim_{n\to\infty} traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = \mu(\sigma)$. From $\mathbf{History}_T(1, W, \varphi, \Gamma)$ by Lem. 76 we know $\mathbf{History}(W, \varphi, \lambda n.traverse(\Gamma, s, n))$. From $\lim_{n\to\infty} traverse(\Gamma, s, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$ and $\forall\sigma. \lim_{n\to\infty} traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = \mu(\sigma)$ we know $W \Downarrow'_\varphi \mu$.

Lemma 78. *For all $P, \mathbb{C}, Q$, if $\models_{A'} \{P\}\mathbb{C}\{Q\}$ then $\models_{AT} \{P\}\mathbb{C}\{Q\}$.*

Proof. For all $P, \mathbb{C}, Q$ such that $\models_{A'} \{P\}\mathbb{C}\{Q\}$, we need to prove for all μ, if $\mu \models P$, then for all φ, s, and μ', if $init(\mathbb{C}, \mu) \Downarrow^s_\varphi \mu'$, then $\mu' \models Q$. For all μ, φ, s, μ' such that $\mu \models P$ and $init(\mathbb{C}, \mu) \Downarrow^s_\varphi \mu'$, by Lem. 77 we know $init(\mathbb{C}, \mu) \Downarrow'_\varphi \mu'$. From $\models_{A'} \{P\}\mathbb{C}\{Q\}$ and $\mu \models P$ we know $\mu' \models Q$.

Definition 47. $\mathbf{RemoveSplit}(W) \stackrel{\text{def}}{=} \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\delta(\mathbf{RemoveSplit}(\mathbb{C}), \sigma)\}$.

Lemma 79. *For all W, $\mathbf{RemoveSplit}(W) = \lambda(\mathbb{C}, \sigma). \sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot W(\mathbb{C}', \sigma)$.*

Proof. For all W, we have

$$\begin{aligned}
&\mathbf{RemoveSplit}(W)\\
&= \mathbb{E}_{(\mathbb{C}',\sigma')\sim W}\{\delta(\mathbf{RemoveSplit}(\mathbb{C}'), \sigma')\}\\
&= \lambda(\mathbb{C}, \sigma). \textstyle\sum_{\mathbb{C}',\sigma'} W(\mathbb{C}', \sigma') \cdot \delta(\mathbf{RemoveSplit}(\mathbb{C}'), \sigma')(\mathbb{C}, \sigma)\\
&= \lambda(\mathbb{C}, \sigma). \textstyle\sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot W(\mathbb{C}', \sigma).
\end{aligned}$$

Lemma 80. *For all $\mathbb{C}$ and μ, $\mathbf{RemoveSplit}(init(\mathbb{C}, \mu)) = init(\mathbf{RemoveSplit}(\mathbb{C}), \mu)$.*

Proof. For all $\mathbb{C}$ and μ,

$$\begin{aligned}
&\mathbf{RemoveSplit}(init(\mathbb{C}, \mu))\\
&= \mathbf{RemoveSplit}(\delta(\mathbb{C}) \otimes \mu)\\
&= \lambda(\mathbb{C}', \sigma'). \textstyle\sum_{\mathbb{C}''} \delta(\mathbf{RemoveSplit}(\mathbb{C}''))(\mathbb{C}') \cdot (\delta(\mathbb{C}) \otimes \mu)(\mathbb{C}'', \sigma') \qquad \text{(by Lem. 79)}\\
&= \lambda(\mathbb{C}', \sigma'). \textstyle\sum_{\mathbb{C}''} \delta(\mathbf{RemoveSplit}(\mathbb{C}''))(\mathbb{C}') \cdot \delta(\mathbb{C})(\mathbb{C}'') \cdot \mu(\sigma')\\
&= \lambda(\mathbb{C}', \sigma'). \delta(\mathbf{RemoveSplit}(\mathbb{C}))(\mathbb{C}') \cdot \mu(\sigma')\\
&= \delta(\mathbf{RemoveSplit}(\mathbb{C})) \otimes \mu\\
&= init(\mathbf{RemoveSplit}(\mathbb{C}), \mu).
\end{aligned}$$

Lemma 81. *For all p, W, φ, there exists Γ such that $\boldsymbol{History}_T(p, W, \varphi, \Gamma)$.*

Proof. by coinduction. For all p, W, φ, from the definition of *Schedule*, there exists t and φ' such that $\varphi = t :: \varphi'$. By Lem. 61, there exists W' such that $W \overset{t}{\rightsquigarrow} W'$. It is obvious that $\#nextsplit(W, t) = 1$ or $\#nextsplit(W, t) > 1$, we prove the two cases respectively.

- $\#nextsplit(W,t) = 1$.
  There exists $b_1,\ldots,b_k$ such that $nextsplit(W) = \{\mathbf{split}(b_1,\ldots,b_k)\}$ and $\mathbf{validsplit}(\mathbf{split}(b_1,\ldots,b_k))$. By Lem. 65 we know there exists i such that $W'|_{b_i}$ exists. By Lem. 63 we know $[\![\mathbf{Pr}(b_i)]\!]_{W'(State)} > 0$, thus $splitter(W', sp)$ is not empty. It is obvious that $splitter(W', sp)$ is finite, so there exists $W_0, p_0, \ldots, W_k, p_k$ such that
  $splitter(W', sp) = \{(W_0, p_0), \ldots, (W_k, p_k)\}$.
  By coinduction hypothesis there eixsts $\Gamma_0, \ldots, \Gamma_k$ such that $\forall i.\ \mathbf{History}_T(p \cdot p_i, W_i, \varphi', \Gamma_i)$.
  From $W \overset{t}{\rightsquigarrow} W'$, $nextsplit(W,t) = \{sp\}$ and $splitter(W', sp) = \{(W_0, p_0), \ldots, (W_k, p_k)\}$ we have $\mathbf{History}_T(p, W, t :: \varphi', \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k))$. Let $\Gamma \overset{\text{def}}{=} \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k)$, then $\mathbf{History}_T(p, W, \varphi, \Gamma)$.
- $\#nextsplit(W,t) > 1$.
  By coinduction hypothesis there eixsts Γ' such that $\mathbf{History}_T(p, W', \varphi', \Gamma')$.
  From $W \overset{t}{\rightsquigarrow} W'$ and $\#nextsplit(W,t) > 1$ we have $\mathbf{History}_T(p, W, t :: \varphi', \mathbf{Tree}(p, W, \Gamma'))$. Let $\Gamma \overset{\text{def}}{=} \mathbf{Tree}(p, W, \Gamma')$, then $\mathbf{History}_T(p, W, \varphi, \Gamma)$.

Definition 48. Let P be a Prop, $\chi(P) = \begin{cases} 1, & \text{if P holds} \\ 0, & \text{otherwise.} \end{cases}$.

Lemma 82. *For all* $W, b, \mathbb{C}, \sigma$, $W|_b = \lambda(\mathbb{C},\sigma).\frac{\chi(\sigma \models b_i)\cdot W(\mathbb{C},\sigma)}{[\![\mathbf{Pr}(b)]\!]_{W(State)}}$.

Proof. For all $W, b, \mathbb{C}, \sigma$,

$$\begin{aligned}
W|_b &= \lambda(\mathbb{C},\sigma).W|_{\lambda(\mathbb{C},\sigma).\sigma\models b}(\mathbb{C},\sigma) \\
&= \lambda(\mathbb{C},\sigma).\begin{cases} \frac{W(\mathbb{C},\sigma)}{\mathbf{Pr}_{(\mathbb{C},\sigma)\sim W}[\sigma\models b]}, & \text{if } \sigma \models b \\ 0, & \text{otherwise} \end{cases} \\
&= \lambda(\mathbb{C},\sigma).\frac{\chi(\sigma\models b)\cdot W(\mathbb{C},\sigma)}{\mathbf{Pr}_{(\mathbb{C},\sigma)\sim W}[\sigma\models b]} \\
&= \lambda(\mathbb{C},\sigma).\frac{\chi(\sigma\models b)\cdot W(\mathbb{C},\sigma)}{[\![\mathbf{Pr}(b)]\!]_{W(State)}}. \qquad \text{(by Lem. 62)}
\end{aligned}$$

Lemma 83. *For all* W, sp, *if* $\mathbf{validsplit}(sp)$ *and* $splitter(W, sp) = \{(W_0, p_0), \ldots, (W_k, p_k)\}$, *then* $\sum_{i=0}^{k} p_i = 1$ *and for all* $\mathbb{C}$ *and* σ, $W(\mathbb{C},\sigma) = \sum_{i=0}^{k} p_i \cdot W_i(\mathbb{C},\sigma)$.

Proof. For all W, sp such that $\mathbf{validsplit}(sp)$ and $splitter(W, sp) = \{(W_0, p_0), \ldots, (W_k, p_k)\}$, there exists $b_1, \ldots, b_n$ such that $sp = \mathbf{split}(b_1, \ldots, b_n)$. From $\mathbf{validsplit}(sp)$ by Lem. 64 we know $\sum_{i=1}^{n} [\![\mathbf{Pr}(b_i)]\!]_{W(State)} = 1$. From $splitter(W, sp) = \{(W_0, p_0), \ldots, (W_k, p_k)\}$ we know
$\{(W|_{b_i}, [\![\mathbf{Pr}(b_i)]\!]_{W(State)}) \mid 1 \le i \le n \wedge [\![\mathbf{Pr}(b_i)]\!]_{W(State)} > 0\} = \{(W_0, p_0), \ldots, (W_k, p_k)\}$,
thus $\sum_{i=0}^{k} p_i = \sum_i \{[\![\mathbf{Pr}(b_i)]\!]_{W(State)} \mid 1 \le i \le n \wedge [\![\mathbf{Pr}(b_i)]\!]_{W(State)} > 0\} =$

$\sum_{i=1}^{n} [\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}} = 1$. For all $\mathbb{C}$ and σ,

$$
\begin{aligned}
&\sum_{i=0}^{k} p_i \cdot W_i(\mathbb{C},\sigma)\\
&= \sum_i \{[\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}} \cdot W|_{b_i}(\mathbb{C},\sigma) \mid 1 \le i \le n \wedge [\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}} > 0\}\\
&= \sum_i \{[\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}} \cdot \tfrac{\chi(\sigma \models b_i)\cdot W(\mathbb{C},\sigma)}{[\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}}} \mid 1 \le i \le n \wedge [\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}} > 0\} \qquad \text{(by Lem. 82)}\\
&= \sum_i \{\chi(\sigma \models b_i) \cdot W(\mathbb{C},\sigma) \mid 1 \le i \le n \wedge [\![\mathbf{Pr}(b_i)]\!]_{W^{(State)}} > 0\}\\
&= \sum_i \{\chi(\sigma \models b_i) \cdot W(\mathbb{C},\sigma) \mid 1 \le i \le n \wedge \mathbf{Pr}_{(\mathbb{C},\sigma)\sim W}[\sigma \models b_i] > 0\} \qquad \text{(by Lem. 62)}\\
&= \sum_i \{\chi(\sigma \models b_i) \cdot W(\mathbb{C},\sigma) \mid 1 \le i \le n \wedge \textstyle\sum_{\mathbb{C},\sigma}\{W(\mathbb{C},\sigma) \mid \sigma \models b_i\} > 0\}\\
&= \sum_i \{\chi(\sigma \models b_i) \cdot W(\mathbb{C},\sigma) \mid 1 \le i \le n \wedge \exists \mathbb{C},\sigma. W(\mathbb{C},\sigma) > 0 \wedge \sigma \models b_i\}\\
&= \sum_i \{\chi(\sigma \models b_i) \cdot W(\mathbb{C},\sigma) \mid 1 \le i \le n \wedge \exists \mathbb{C},\sigma. \chi(\mathbb{C},\sigma) \cdot W(\mathbb{C},\sigma) > 0\}\\
&= W(\mathbb{C},\sigma) \cdot \textstyle\sum_{i=1}^{n} \chi(\sigma \models b_i)\\
&= W(\mathbb{C},\sigma). \qquad \text{(from } \mathbf{validsplit}(\mathbf{split}(b_1,\ldots,b_k)))
\end{aligned}
$$

Lemma 84. *For all p, W, φ, Γ, if $\boldsymbol{History}_T(p, W, \varphi, \Gamma)$, then $prob(\Gamma, s, 0) = \chi(s = zeros) \cdot p$ and $traverse(\Gamma, s, 0) = W$ for all s.*

Proof. For all p, W, φ, Γ such that $\mathbf{History}_T(p, W, \varphi, \Gamma)$, there are two cases.

- $\varphi = t :: \varphi'$, $\Gamma = \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k)$, $W \overset{t}{\rightsquigarrow} W'$, $nextsplit(W, t) = \{sp\}$, $splitter(W', sp) = \{(W_0, p_0), \ldots, (W_k, p_k)\}$ and $\forall i.\, \mathbf{History}_T(p{\cdot}p_i, W_i, \varphi', \Gamma_i)$.
  For all s, $prob(\Gamma, s, 0) = prob(\mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k), s, 0) = \chi(s = zeros) \cdot p$, and $traverse(\Gamma, s, 0) = traverse(\mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k), s, 0) = W$.
- $\varphi = t :: \varphi'$, $\Gamma = \mathbf{Tree}(p, W, \Gamma')$, $W \overset{t}{\rightsquigarrow} W'$, $\#nextsplit(W, t) > 1$ and $\mathbf{History}_T(p, W', \varphi', \Gamma')$.
  For all s, $prob(\Gamma, s, 0) = prob(\mathbf{Tree}(p, W, \Gamma'), s, 0) = \chi(s = zeros) \cdot p$, and $traverse(\Gamma, s, 0) = traverse(\mathbf{Tree}(p, W, \Gamma'), s, 0) = W$.

Lemma 85. *For all W, W', t, if $W \overset{t}{\rightsquigarrow} W'$, then $\mathbf{RemoveSplit}(W) \overset{t}{\rightsquigarrow} \mathbf{RemoveSplit}(W')$.*

Proof. For all W, W', t such that $W \overset{t}{\rightsquigarrow} W'$, we have

$$
\begin{aligned}
&\lambda(\mathbb{C}',\sigma'). \textstyle\sum_{\mathbb{C},\sigma}\{\mathbf{RemoveSplit}(W)(\mathbb{C},\sigma) \cdot p \mid (\mathbb{C},\sigma) \xrightarrow[t]{p} (\mathbb{C}',\sigma')\}\\
&= \lambda(\mathbb{C}',\sigma'). \textstyle\sum_{\mathbb{C},\sigma}\{\sum_{\mathbb{C}''} \delta(\mathbf{RemoveSplit}(\mathbb{C}''))(\mathbb{C}) \cdot W(\mathbb{C}'',\sigma) \cdot p \mid (\mathbb{C},\sigma) \xrightarrow[t]{p} (\mathbb{C}',\sigma')\} \qquad \text{(by Lem. 79)}\\
&= \lambda(\mathbb{C}',\sigma'). \textstyle\sum_{\mathbb{C},\sigma,\mathbb{C}''}\{\delta(\mathbf{RemoveSplit}(\mathbb{C}''))(\mathbb{C}) \cdot W(\mathbb{C}'',\sigma) \cdot p \mid (\mathbb{C},\sigma) \xrightarrow[t]{p} (\mathbb{C}',\sigma')\}\\
&= \lambda(\mathbb{C}',\sigma'). \textstyle\sum_{\mathbb{C}'',\sigma}\{W(\mathbb{C}'',\sigma) \cdot p \mid (\mathbf{RemoveSplit}(\mathbb{C}''),\sigma) \xrightarrow[t]{p} (\mathbb{C}',\sigma')\}\\
&= \lambda(\mathbb{C}',\sigma'). \textstyle\sum_{\mathbb{C}'',\sigma}\{W(\mathbb{C}'',\sigma) \cdot p \mid \exists!\mathbb{C}.\ (\mathbb{C}'',\sigma) \xrightarrow[t]{p} (\mathbb{C},\sigma') \wedge \mathbb{C}' = \mathbf{RemoveSplit}(\mathbb{C})\} \qquad \text{(by Lem. 71)}\\
&= \lambda(\mathbb{C}',\sigma'). \textstyle\sum_{\mathbb{C},\sigma,\mathbb{C}''}\{W(\mathbb{C}'',\sigma) \cdot p \mid (\mathbb{C}'',\sigma) \xrightarrow[t]{p} (\mathbb{C},\sigma') \wedge \mathbb{C}' = \mathbf{RemoveSplit}(\mathbb{C})\}\\
&= \lambda(\mathbb{C}',\sigma'). \textstyle\sum_{\mathbb{C},\sigma,\mathbb{C}''}\{\delta(\mathbf{RemoveSplit}(\mathbb{C}))(\mathbb{C}') \cdot W(\mathbb{C}'',\sigma) \cdot p \mid (\mathbb{C}'',\sigma) \xrightarrow[t]{p} (\mathbb{C},\sigma')\}\\
&= \lambda(\mathbb{C}',\sigma'). \textstyle\sum_{\mathbb{C}} \delta(\mathbf{RemoveSplit}(\mathbb{C}))(\mathbb{C}') \cdot \sum_{\mathbb{C}'',\sigma}\{W(\mathbb{C}'',\sigma) \cdot p \mid (\mathbb{C}'',\sigma) \xrightarrow[t]{p} (\mathbb{C},\sigma')\}\\
&= \lambda(\mathbb{C}',\sigma'). \textstyle\sum_{\mathbb{C}} \delta(\mathbf{RemoveSplit}(\mathbb{C}))(\mathbb{C}') \cdot W'(\mathbb{C},\sigma') \qquad (\text{by } W \overset{t}{\rightsquigarrow} W')\\
&= \mathbf{RemoveSplit}(W'). \qquad \text{(by Lem. 79)}
\end{aligned}
$$

Therefore, $\mathbf{RemoveSplit}(W) \overset{t}{\rightsquigarrow} \mathbf{RemoveSplit}(W')$.

Lemma 86. *For all* $n, W, W_0, \ldots, W_k, p_0, \ldots, p_k, t, W', W'_0, \ldots, W'_k$, *if* $W = \lambda(\mathbb{C}, \sigma). \sum_{i=0}^{k} p_i \cdot W_i(\mathbb{C}, \sigma)$, $W \overset{t}{\rightsquigarrow} W'$ *and* $\forall i.\ W_i \overset{t}{\rightsquigarrow} W_i$, *then* $W' = \lambda(\mathbb{C}, \sigma). \sum_{i=0}^{k} p_i \cdot W'_i(\mathbb{C}, \sigma)$.

Proof. For all $n, W, W_0, \ldots, W_k, p_0, \ldots, p_k, t, W', W'_0, \ldots, W'_k$ sucht that $W = \lambda(\mathbb{C}, \sigma). \sum_{i=0}^{k} p_i \cdot W_i(\mathbb{C}, \sigma)$, $W \overset{t}{\rightsquigarrow} W'$ and $\forall i.\ W_i \overset{t}{\rightsquigarrow} W_i$, we have

$$
\begin{aligned}
W' &= \lambda(\mathbb{C}', \sigma'). \textstyle\sum_{\mathbb{C},\sigma}\{W(\mathbb{C}, \sigma) \cdot p \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')\} \\
&= \lambda(\mathbb{C}', \sigma'). \textstyle\sum_{\mathbb{C},\sigma}\{\sum_{i=0}^{k} p_i \cdot W_i(\mathbb{C}, \sigma) \cdot p \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')\} \\
&= \lambda(\mathbb{C}', \sigma'). \textstyle\sum_{i=0}^{k} p_i \cdot \sum_{\mathbb{C},\sigma}\{W_i(\mathbb{C}, \sigma) \cdot p \mid (\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma')\} \\
&= \lambda(\mathbb{C}', \sigma'). \textstyle\sum_{i=0}^{k} p_i \cdot W'_i(\mathbb{C}', \sigma') \\
&= \lambda(\mathbb{C}, \sigma). \textstyle\sum_{i=0}^{k} p_i \cdot W'_i(\mathbb{C}, \sigma).
\end{aligned}
$$

Lemma 87. *For all* $n, W, W_0, \ldots, W_k, p_0, \ldots, p_k, \varphi, \vec{W}, \vec{W}_0, \ldots, \vec{W}_k$, *if* $\boldsymbol{History}(W, \varphi, \vec{W})$, $W = \lambda(\mathbb{C}, \sigma). \sum_{i=0}^{k} p_i \cdot W_i(\mathbb{C}, \sigma)$, $\mathbf{Nosplit}(W)$, $\forall i.\ \boldsymbol{History}(W_i, \varphi, \vec{W}_i)$ *and* $\forall i.\ \mathbf{Nosplit}(W_i)$, *then* $\vec{W}[n] = \lambda(\mathbb{C}, \sigma). \sum_{i=0}^{k} p_i \cdot \vec{W}_i[n](\mathbb{C}, \sigma)$.

Proof. by induction on n.

- base case: $n = 0$.
  For all $W, W_0, \ldots, W_k, p_0, \ldots, p_k, \varphi, \vec{W}, \vec{W}_0, \ldots, \vec{W}_k$ such that $\mathbf{History}(W, \varphi, \vec{W})$, $\mathbf{Nosplit}(W)$, $W = \lambda(\mathbb{C}, \sigma). \sum_{i=0}^{k} p_i \cdot W_i(\mathbb{C}, \sigma)$, $\forall i.\ \mathbf{History}(W_i, \varphi, \vec{W}_i)$ and $\forall i.\ \mathbf{Nosplit}(W_i)$,
  from $\mathbf{History}(W, \varphi, \vec{W})$ by Lem. 50 we know $\vec{W}[0] = W$. For all i, from $\mathbf{History}(W_i, \varphi, \vec{W}_i)$ by Lem. 50 we know $\vec{W}_i[0] = W_i$. From $W = \lambda(\mathbb{C}, \sigma). \sum_{i=0}^{k} p_i \cdot W_i(\mathbb{C}, \sigma)$ we know $\vec{W}[0] = \lambda(\mathbb{C}, \sigma). \sum_{i=0}^{k} p_i \cdot \vec{W}_i[0](\mathbb{C}, \sigma)$.
- inductive case: $n = n' + 1$.
  IH: for all $W, W_0, \ldots, W_k, p_0, \ldots, p_k, \varphi, \vec{W}, \vec{W}_0, \ldots, \vec{W}_k$, if $\mathbf{History}(W, \varphi, \vec{W})$, $\mathbf{Nosplit}(W)$, $W = \lambda(\mathbb{C}, \sigma). \sum_{i=0}^{k} p_i \cdot W_i(\mathbb{C}, \sigma)$, $\forall i.\ \mathbf{History}(W_i, \varphi, \vec{W}_i)$ and $\forall i.\ \mathbf{Nosplit}(W_i)$, then $\vec{W}[n'] = \lambda(\mathbb{C}, \sigma). \sum_{i=0}^{k} p_i \cdot \vec{W}_i[n'](\mathbb{C}, \sigma)$.
  For all $W, W_0, \ldots, W_k, p_0, \ldots, p_k, \varphi, \vec{W}, \vec{W}_0, \ldots, \vec{W}_k$ such that $\mathbf{History}(W, \varphi, \vec{W})$, $\mathbf{Nosplit}(W)$, $W = \lambda(\mathbb{C}, \sigma). \sum_{i=0}^{k} p_i \cdot W_i(\mathbb{C}, \sigma)$, $\forall i.\ \mathbf{History}(W_i, \varphi, \vec{W}_i)$, and $\forall i.\ \mathbf{Nosplit}(W_i)$,
  from $\mathbf{History}(W, \varphi, \vec{W})$ we know there exists $t, \varphi', W', \vec{W}'$ such that $\varphi = t :: \varphi'$, $W \overset{t}{\hookrightarrow} W'$, $\mathbf{History}(W', \varphi', \vec{W}')$ and $\vec{W} = W :: \vec{W}'$. From $\mathbf{Nosplit}(W)$ and $W \overset{t}{\hookrightarrow} W'$ by Lem. 49 we know $W \overset{t}{\rightsquigarrow} W'$, by Lem. 53 we know $\mathbf{Nosplit}(W')$. For all i, from $\mathbf{History}(W_i, \varphi, \vec{W}_i)$ and $\varphi = t :: \varphi'$ we know

there exists $W_i', \vec{W}_i'$ such that $W_i \overset{t}{\hookrightarrow} W_i'$, $\mathbf{History}(W_i', \varphi', \vec{W}_i')$ and $\vec{W}_i = W_i :: \vec{W}_i'$. From $\mathbf{Nosplit}(W_i)$ and $W_i \overset{t}{\hookrightarrow} W_i'$ by Lem. 49 we know $W_i \overset{t}{\rightsquigarrow} W_i'$, by Lem. 53 we know $\mathbf{Nosplit}(W_i')$. From $W = \lambda(\mathbb{C},\sigma).\sum_{i=0}^{k} p_i \cdot W_i(\mathbb{C},\sigma)$, $W \overset{t}{\rightsquigarrow} W'$ and $\forall i.\ W_i \overset{t}{\rightsquigarrow} W_i'$ by Lem. 86 we know $W' = \lambda(\mathbb{C},\sigma).\sum_{i=0}^{k} p_i \cdot W_i'(\mathbb{C},\sigma)$. From $\mathbf{History}(W', \varphi', \vec{W}')$, $\mathbf{Nosplit}(W')$, $\forall i.\ \mathbf{History}(W_i', \varphi', \vec{W}_i')$ and $\forall i.\ \mathbf{Nosplit}(W_i')$ by IH we know $\vec{W}'[n'] = \lambda(\mathbb{C},\sigma).\sum_{i=0}^{k} p_i \cdot \vec{W}_i'[n'](\mathbb{C},\sigma)$, thus $\vec{W}[n] = (W :: \vec{W}')[n'+1] = \vec{W}'[n] = \lambda(\mathbb{C},\sigma).\sum_{i=0}^{k} p_i \cdot \vec{W}_i'[n'](\mathbb{C},\sigma) = \lambda(\mathbb{C},\sigma).\sum_{i=0}^{k} p_i \cdot (W_i :: \vec{W}_i')[n'+1](\mathbb{C},\sigma) = \lambda(\mathbb{C},\sigma).\sum_{i=0}^{k} p_i \cdot (\vec{W}_i)[n](\mathbb{C},\sigma)$.

Lemma 88. *For all* $n, W, \varphi, \vec{W}, \Gamma, p$, *if* $\boldsymbol{History}(\mathbf{RemoveSplit}(W), \varphi, \vec{W})$ *and* $\boldsymbol{History}_T(p, W, \varphi, \Gamma)$, *then* $\vec{W}[n](\mathbb{C},\sigma)\cdot p = \sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot \sum_s prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbb{C}', \sigma)$ *for all* $\mathbb{C}$ *and* σ.

Proof. by induction on n.

- base case: $n = 0$.
  For all $W, \varphi, \vec{W}, \Gamma, p$ such that $\mathbf{History}(\mathbf{RemoveSplit}(W), \varphi, \vec{W})$ and $\mathbf{History}_T(p, W, \varphi, \Gamma)$, from $\mathbf{History}(\mathbf{RemoveSplit}(W), \varphi, \vec{W})$ by Lem. 50 we know $\vec{W}[0] = \mathbf{RemoveSplit}(W)$. From $\mathbf{History}_T(p, W, \varphi, \Gamma)$ by Lem. 84 we know for all s, $traverse(\Gamma, s, 0) = W$ and $prob(\Gamma, s, 0) = \chi(s = zeros) \cdot p$. For all $\mathbb{C}$ and σ, we have

  $$\begin{aligned}
  &\vec{W}[0](\mathbb{C},\sigma) \cdot p \\
  &= \mathbf{RemoveSplit}(W)(\mathbb{C},\sigma) \cdot p \\
  &= \mathbb{E}_{(\mathbb{C}',\sigma')\sim W}\{\delta(\mathbf{RemoveSplit}(\mathbb{C}'), \sigma')\}(\mathbb{C},\sigma) \cdot p \\
  &= \textstyle\sum_{\mathbb{C}',\sigma'} W(\mathbb{C}',\sigma') \cdot \delta(\mathbf{RemoveSplit}(\mathbb{C}'), \sigma')(\mathbb{C},\sigma) \cdot p \\
  &= \textstyle\sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot p \cdot W(\mathbb{C}',\sigma) \\
  &= \textstyle\sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot \chi(zeros = zeros) \cdot p \cdot traverse(\Gamma, zeros, 0)(\mathbb{C}',\sigma) \\
  &= \textstyle\sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot \sum_s \chi(s = zeros) \cdot p \cdot traverse(\Gamma, s, 0)(\mathbb{C}',\sigma) \\
  &= \textstyle\sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot \sum_s prob(\Gamma, s, 0) \cdot traverse(\Gamma, s, 0)(\mathbb{C}',\sigma).
  \end{aligned}$$

- inductive case: $n = n' + 1$.
  IH: for all $W, \varphi, \vec{W}, \Gamma, p$, if $\mathbf{History}(\mathbf{RemoveSplit}(W), \varphi, \vec{W})$ and $\mathbf{History}_T(p, W, \varphi, \Gamma)$, then $\vec{W}[n'](\mathbb{C},\sigma)\cdot p = \sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot \sum_s prob(\Gamma, s, n') \cdot traverse(\Gamma, s, n')(\mathbb{C}', \sigma)$ for all $\mathbb{C}$ and σ.
  For all $W, \varphi, \vec{W}, \Gamma, p$ such that $\mathbf{History}(\mathbf{RemoveSplit}(W), \varphi, \vec{W})$ and $\mathbf{History}_T(p, W, \varphi, \Gamma)$, from $\mathbf{History}(\mathbf{RemoveSplit}(W), \varphi, \vec{W})$ we know there exists t, φ', W'' such that $\varphi = t :: \varphi'$, $\mathbf{RemoveSplit}(W) \overset{t}{\hookrightarrow} W''$, $\mathbf{History}(W'', \varphi', \vec{W}')$ and $\vec{W} = \mathbf{RemoveSplit}(W) :: \vec{W}'$. It is obvious that $\mathbf{Nosplit}(\mathbf{RemoveSplit}(W))$, from $\mathbf{RemoveSplit}(W) \overset{t}{\hookrightarrow} W''$ by Lem. 49 we know $\mathbf{RemoveSplit}(W) \overset{t}{\rightsquigarrow} W''$. From $\mathbf{History}_T(p, W, \varphi, \Gamma)$, there are two cases.

- $\Gamma = \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k)$, $W \overset{t}{\rightsquigarrow} W'$, $\mathit{nextsplit}(W, t) = \{sp\}$, $\mathit{splitter}(W', sp) = \{(W_0, p_0), \ldots, (W_k, p_k)\}$ and $\forall i.\ \mathbf{History}_T(p \cdot p_i, W_i, \varphi', \Gamma_i)$.
  From $W \overset{t}{\rightsquigarrow} W'$ by Lem. 85 we know $\mathbf{RemoveSplit}(W) \overset{t}{\rightsquigarrow} \mathbf{RemoveSplit}(W')$.
  From $\mathbf{RemoveSplit}(W) \overset{t}{\rightsquigarrow} W''$ by Lem. 57 we know $W'' = \mathbf{RemoveSplit}(W')$.
  From $\mathbf{History}(W'', \varphi', \vec{W}')$ we know $\mathbf{History}(\mathbf{RemoveSplit}(W'), \varphi', \vec{W}')$.
  From $\mathit{nextsplit}(W, t) = \{sp\}$ we know $\mathbf{validsplit}(sp)$. From $\mathit{splitter}(W', sp) = \{(W_0, p_0), \ldots, (W_k, p_k)\}$ by Lem. 83 we know $W' = \lambda(\mathbb{C}, \sigma).\ \sum_{i=0}^{k} p_i \cdot W_i(\mathbb{C}, \sigma)$, thus

$$
\begin{aligned}
&\mathbf{RemoveSplit}(W') \\
&= \lambda(\mathbb{C}, \sigma).\ \textstyle\sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot W'(\mathbb{C}', \sigma) \qquad \text{(by Lem. 79)} \\
&= \lambda(\mathbb{C}, \sigma).\ \textstyle\sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot \sum_{i=0}^{k} p_i \cdot W_i(\mathbb{C}', \sigma) \\
&= \lambda(\mathbb{C}, \sigma).\ \textstyle\sum_{i=0}^{k} p_i \cdot \sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot W_i(\mathbb{C}', \sigma) \\
&= \lambda(\mathbb{C}, \sigma).\ \textstyle\sum_{i=0}^{k} p_i \cdot \mathbf{RemoveSplit}(W_i)(\mathbb{C}, \sigma). \qquad \text{(by Lem. 79)}
\end{aligned}
$$

  By Lem. 67 we know for all i, there exists $\vec{W}_i$ such that $\mathbf{History}(\mathbf{RemoveSplit}(W_i), \varphi', \vec{W}_i)$.
  From $W' = \lambda(\mathbb{C}, \sigma).\ \sum_{i=0}^{k} p_i \cdot W_i(\mathbb{C}, \sigma)$, $\mathbf{History}(\mathbf{RemoveSplit}(W'), \varphi', \vec{W}')$, $\mathbf{Nosplit}(\mathbf{RemoveSplit}(W'))$ and $\forall i.\ \mathbf{Nosplit}(\mathbf{RemoveSplit}(W_i))$ by Lem. 87 we know
  $\vec{W}'[n] = \lambda(\mathbb{C}, \sigma).\ \sum_{i=0}^{k} p_i \cdot \vec{W}_i[n](\mathbb{C}, \sigma)$.
  For all i, from $\mathbf{History}(\mathbf{RemoveSplit}(W_i), \varphi', \vec{W}_i)$ and $\mathbf{History}_T(p \cdot p_i, W_i, \varphi', \Gamma_i)$ by IH we know for all $\mathbb{C}$ and σ, $\vec{W}_i[n'](\mathbb{C}, \sigma) \cdot p \cdot p_i = \sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot \sum_s \mathit{prob}(\Gamma_i, s, n') \cdot \mathit{traverse}(\Gamma_i, s, n')(\mathbb{C}', \sigma)$.
  For all $\mathbb{C}$ and σ, we have

$$
\begin{aligned}
&\vec{W}[n](\mathbb{C}, \sigma) \cdot p \\
&= (\mathbf{RemoveSplit}(W) :: \vec{W}')[n' + 1](\mathbb{C}, \sigma) \cdot p \\
&= \vec{W}'[n'](\mathbb{C}, \sigma) \cdot p \\
&= \textstyle\sum_{i=0}^{k} p_i \cdot \vec{W}_i[n'](\mathbb{C}, \sigma) \cdot p \\
&= \textstyle\sum_{i=0}^{k} \sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot \sum_s \mathit{prob}(\Gamma_i, s, n') \cdot \mathit{traverse}(\Gamma_i, s, n')(\mathbb{C}', \sigma) \\
&= \textstyle\sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot \sum_{i=0}^{k} \sum_s \mathit{prob}(\Gamma_i, s, n') \cdot \mathit{traverse}(\Gamma_i, s, n')(\mathbb{C}', \sigma) \\
&= \textstyle\sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot \sum_i \sum_s \mathit{prob}(\Gamma, i :: s, n' + 1) \cdot \mathit{traverse}(\Gamma, i :: s, n' + 1)(\mathbb{C}', \sigma) \\
&= \textstyle\sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot \sum_s \mathit{prob}(\Gamma, s, n) \cdot \mathit{traverse}(\Gamma, s, n)(\mathbb{C}', \sigma).
\end{aligned}
$$

- $\Gamma = \mathbf{Tree}(p, W, \Gamma')$, $W \overset{t}{\rightsquigarrow} W'$, $\#\mathit{nextsplit}(W, t) > 1$ and $\mathbf{History}_T(p, W', \varphi', \Gamma')$.
  From $W \overset{t}{\rightsquigarrow} W'$ by Lem. 85 we know $\mathbf{RemoveSplit}(W) \overset{t}{\rightsquigarrow} \mathbf{RemoveSplit}(W')$.
  From $\mathbf{RemoveSplit}(W) \overset{t}{\rightsquigarrow} W''$ by Lem. 57 we know $W'' = \mathbf{RemoveSplit}(W')$.

From $\mathbf{History}(W'', \varphi', \overrightarrow{W}')$ we know $\mathbf{History}(\mathbf{RemoveSplit}(W'), \varphi', \overrightarrow{W}')$.

For all $\mathbb{C}$ and σ, From $\mathbf{History}(\mathbf{RemoveSplit}(W'), \varphi', \overrightarrow{W}')$ and $\mathbf{History}_T(p, W', \varphi', \Gamma')$ by IH we know
$\overrightarrow{W}'[n'](\mathbb{C}, \sigma) \cdot p = \sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot \sum_s prob(\Gamma', s, n') \cdot traverse(\Gamma', s, n')(\mathbb{C}', \sigma)$,
thus

$$
\begin{aligned}
&\overrightarrow{W}[n](\mathbb{C}, \sigma) \cdot p \\
&= (\mathbf{RemoveSplit}(W) :: \overrightarrow{W}')[n'+1](\mathbb{C}, \sigma) \cdot p \\
&= \overrightarrow{W}'[n'](\mathbb{C}, \sigma) \cdot p \\
&= \textstyle\sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot \sum_s prob(\Gamma', s, n') \cdot traverse(\Gamma', s, n')(\mathbb{C}', \sigma) \\
&= \textstyle\sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot \sum_s prob(\Gamma, 0 :: s, n'+1) \cdot traverse(\Gamma, 0 :: s, n'+1)(\mathbb{C}', \sigma) \\
&= \textstyle\sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbb{C}) \cdot \sum_s prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbb{C}', \sigma).
\end{aligned}
$$

Lemma 89. *For all $n, p, W, \varphi, \Gamma, s, \sigma$, if $\boldsymbol{History}_T(p, W, \varphi, \Gamma)$, then $prob(\Gamma, s, n+1) \cdot traverse(\Gamma, s, n+1)(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}, \sigma) \geq prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}, \sigma)$.*

Proof. by induction on n.

- base case: $n = 0$.
  For all $p, W, \varphi, \Gamma, s, \sigma$ such that $\mathbf{History}_T(p, W, \varphi, \Gamma)$, there are two cases.
  - $\varphi = t :: \varphi'$, $\Gamma = \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k)$, $W \overset{t}{\rightsquigarrow} W'$, $nextsplit(W, t) = \{sp\}$, $splitter(W', sp) = \{(W_0, p_0), \ldots, (W_k, p_k)\}$ and $\forall i$. $\mathbf{History}_T(p \cdot p_i, W_i, \varphi', \Gamma_i)$.
    $traverse(\Gamma, s, n) = traverse(\mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k), s, 0) = W$.
    $prob(\Gamma, s, n) = prob(\mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k), s, 0) = \chi(s = zeros) \cdot p$.
    If $s \neq zeros$, then $prob(\Gamma, s, n) = 0$, so $prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = 0 \leq prob(\Gamma, s, n+1) \cdot traverse(\Gamma, s, n+1)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$.
    Otherwise $s = zeros$, then $prob(\Gamma, s, n) = p$. From $\mathbf{History}_T(p \cdot p_0, W_0, \varphi', \Gamma_0)$ by Lem. 84 we know $prob(\Gamma_0, zeros, 0) = \chi(zeros = zeros) \cdot p \cdot p_0 = p \cdot p_0$ and $traverse(\Gamma_0, zeros, 0) = W_0$, thus $traverse(\Gamma, s, n+1) = traverse(\mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k), 0 :: zeros, 1) = traverse(\Gamma_0, zeros, 0) = W_0$.
    $prob(\Gamma, s, n+1) = prob(\mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k), 0 :: zeros, 1) = prob(\Gamma_0, zeros, 0) = p \cdot p_0$.
    It is obvious that $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 0$ or $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$, we prove the two cases respectively.
    - $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 0$.
      From $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 0$ we know $\sum_\sigma W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = 0$, thus $W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = 0$ for all σ. Therefore $prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = prob(\Gamma, s, n) \cdot W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = 0 \leq prob(\Gamma, s, n+1) \cdot traverse(\Gamma, s, n+1)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$.

  * $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$.
    From $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$ by Lem. 38 we know $nextsplit(W) \supseteq \{\mathbf{split}(\text{true})\}$. From $nextsplit(W,t) = \{sp\}$ we know $sp = \mathbf{split}(\text{true})$. From $splitter(W', sp) = \{(W_0, p_0), \ldots, (W_k, p_k)\}$ we know $k = 0$, $W_0 = W'|_{\text{true}} = W'$ and $p_0 = [\![\mathbf{Pr}(\text{true})]\!]_{W'^{(State)}} = 1$. From $W \overset{t}{\rightsquigarrow} W'$ by Lem. 31 we know $W'(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \geq W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$, thus $prob(\Gamma, s, n+1) \cdot traverse(\Gamma, s, n+1)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = p \cdot p_0 \cdot W_0(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = p \cdot W'(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \geq p \cdot W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$.

- $\varphi = t :: \varphi'$, $\Gamma = \mathbf{Tree}(p, W, \Gamma')$, $W \overset{t}{\rightsquigarrow} W'$, $\#nextsplit(W,t) > 1$ and $\mathbf{History}_T(p, W', \varphi', \Gamma')$.
  $traverse(\Gamma, s, n) = traverse(\mathbf{Tree}(p, W, \Gamma'), s, 0) = W$.
  $prob(\Gamma, s, n) = prob(\mathbf{Tree}(p, W, \Gamma'), s, 0) = \chi(s = zeros) \cdot p$.
  If $s \neq zeros$, then $prob(\Gamma, s, n) = 0$, so $prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = 0 \leq prob(\Gamma, s, n+1) \cdot traverse(\Gamma, s, n+1)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$.
  Otherwise $s = zeros$, then $prob(\Gamma, s, n) = p$. From $\mathbf{History}_T(p, W', \varphi', \Gamma')$ by Lem. 84 we know $prob(\Gamma', zeros, 0) = \chi(zeros = zeros) \cdot p = p$ and $traverse(\Gamma', zeros, 0) = W'$, so
  $traverse(\Gamma, s, n{+}1) = traverse(\mathbf{Tree}(p, W, \Gamma'), 0 :: zeros, 1) = traverse(\Gamma', zeros, 0) = W'$.
  $prob(\Gamma, s, n{+}1) = prob(\mathbf{Tree}(p, W, \Gamma'), 0 :: zeros, 1) = prob(\Gamma', zeros, 0) = p$.
  From $W \overset{t}{\rightsquigarrow} W'$ by Lem. 31 we know $W'(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \geq W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$, thus $prob(\Gamma, s, n{+}1) \cdot traverse(\Gamma, s, n{+}1)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = p \cdot W'(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \geq p \cdot W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$.

– inductive case: $n = n' + 1$.
  IH: for all $p, W, \varphi, \Gamma, s, \sigma$, if $\mathbf{History}_T(p, W, \varphi, \Gamma)$, then $prob(\Gamma, s, n'+1) \cdot traverse(\Gamma, s, n'{+}1)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \geq prob(\Gamma, s, n') \cdot traverse(\Gamma, s, n')(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$.
  For all $p, W, \varphi, \Gamma, s, \sigma$ such that $\mathbf{History}_T(p, W, \varphi, \Gamma)$, there are two cases.

  - $\varphi = t :: \varphi'$, $\Gamma = \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k)$, $W \overset{t}{\rightsquigarrow} W'$, $nextsplit(W,t) = \{sp\}$, $splitter(W', sp) = \{(W_0, p_0), \ldots, (W_k, p_k)\}$ and $\forall i.\ \mathbf{History}_T(p \cdot p_i, W_i, \varphi', \Gamma_i)$.
    There exists m and s' such that $s = m :: s'$.
    $traverse(\Gamma, s, n) = traverse(\mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k), m :: s', n'+1) = traverse(\Gamma_m, s, n')$.
    $prob(\Gamma, s, n) = prob(\mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k), m :: s', n'{+}1) = prob(\Gamma_m, s', n')$.
    $traverse(\Gamma, s, n+1) = traverse(\mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k), m :: s', n'+2) = traverse(\Gamma_m, s, n'+1)$.
    $prob(\Gamma, s, n{+}1) = prob(\mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k), m :: s', n'{+}2) = prob(\Gamma_m, s', n'{+}1)$.

From $\mathbf{History}_T(p \cdot p_m, W_m, \varphi', \Gamma_m)$ by IH we know $prob(\Gamma_m, s', n'+1) \cdot traverse(\Gamma_m, s', n'+1)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \geq prob(\Gamma_m, s', n') \cdot traverse(\Gamma_m, s', n')(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$, thus $prob(\Gamma, s, n{+}1) \cdot traverse(\Gamma, s, n{+}1)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \geq prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$.

- $\varphi = t :: \varphi'$, $\Gamma = \mathbf{Tree}(p, W, \Gamma')$, $W \overset{t}{\rightsquigarrow} W'$, $\#nextsplit(W, t) > 1$ and $\mathbf{History}_T(p, W', \varphi', \Gamma')$.
  There exists m and s' such that $s = m :: s'$.
  $traverse(\Gamma, s, n) = traverse(\mathbf{Tree}(p, W, \Gamma'), m :: s', n'{+}1) = traverse(\Gamma', s, n')$.
  $prob(\Gamma, s, n) = prob(\mathbf{Tree}(p, W, \Gamma'), m :: s', n'+1) = prob(\Gamma', s', n')$.
  $traverse(\Gamma, s, n{+}1) = traverse(\mathbf{Tree}(p, W, \Gamma'), m :: s', n'{+}2) = traverse(\Gamma', s, n'{+}1)$.
  $prob(\Gamma, s, n+1) = prob(\mathbf{Tree}(p, W, \Gamma'), m :: s', n'+2) = prob(\Gamma', s', n'+1)$.
  From $\mathbf{History}_T(p, W', \varphi', \Gamma')$ by IH we know $prob(\Gamma', s', n'{+}1) \cdot traverse(\Gamma', s', n'{+}1)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \geq prob(\Gamma', s', n') \cdot traverse(\Gamma', s', n')(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$, thus $prob(\Gamma, s, n{+}1) \cdot traverse(\Gamma, s, n{+}1)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \geq prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$.

Lemma 90. *For all n, p, W, φ, Γ, if $\boldsymbol{History}_T(p, W, \varphi, \Gamma)$, then $\sum_s prob(\Gamma, s, n) = p$.*

Proof. By induction on n.

- base case: $n = 0$.
  For all p, W, φ, Γ such that $\mathbf{History}_T(p, W, \varphi, \Gamma)$, there are two cases.
  - $\varphi = t :: \varphi'$, $\Gamma = \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k)$, $W \overset{t}{\rightsquigarrow} W'$, $nextsplit(W, t) = \{sp\}$, $splitter(W', sp) = \{(W_0, p_0), \ldots, (W_k, p_k)\}$ and $\forall i.\ \mathbf{History}_T(p \cdot p_i, W_i, \varphi', \Gamma_i)$.
    $\sum_s prob(\Gamma, s, 0) = \sum_s prob(\mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k), s, 0) = \sum_s \chi(s = zeros) \cdot p = p$.
  - $\varphi = t :: \varphi'$, $\Gamma = \mathbf{Tree}(p, W, \Gamma')$, $W \overset{t}{\rightsquigarrow} W'$, $\#nextsplit(W, t) > 1$ and $\mathbf{History}_T(p, W', \varphi', \Gamma')$.
    $\sum_s prob(\Gamma, s, 0) = \sum_s prob(\mathbf{Tree}(p, W, \Gamma'), s, 0) = \sum_s \chi(s = zeros) \cdot p = p$.
- inductive case: $n = n' + 1$.
  IH: for all p, W, φ, Γ, if $\mathbf{History}_T(p, W, \varphi, \Gamma)$, then $\sum_s prob(\Gamma, s, n') = p$.
  For all p, W, φ, Γ such that $\mathbf{History}_T(p, W, \varphi, \Gamma)$, there are two cases.
  - $\varphi = t :: \varphi'$, $\Gamma = \mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k)$, $W \overset{t}{\rightsquigarrow} W'$, $nextsplit(W, t) = \{sp\}$, $splitter(W', sp) = \{(W_0, p_0), \ldots, (W_k, p_k)\}$ and $\forall i.\ \mathbf{History}_T(p \cdot p_i, W_i, \varphi', \Gamma_i)$.
    From $\forall i.\ \mathbf{History}_T(p{\cdot}p_i, W_i, \varphi', \Gamma_i)$ by IH we have $\forall i.\ \sum_{s'} prob(\Gamma_i, s', n') = p \cdot p_i$.
    From $nextsplit(W, t) = \{sp\}$ we know $\mathbf{validsplit}(sp)$. From $\mathbf{validsplit}(sp)$

and $\mathit{splitter}(W', sp)$
$= \{(W_0, p_0), \ldots, (W_k, p_k)\}$ by Lem. 83 we know $\sum_{i=0}^{k} p_i = 1$, so

$$\begin{aligned}
&\textstyle\sum_s \mathit{prob}(\Gamma, s, n)\\
=\ &\textstyle\sum_{i,s'}\{\mathit{prob}(\mathbf{Tree}(p, W, \Gamma_0, \ldots, \Gamma_k), i :: s', n'+1) \mid i \le k\}\\
=\ &\textstyle\sum_{i=0}^{k}\sum_{s'} \mathit{prob}(\Gamma_i, s', n')\\
=\ &\textstyle\sum_{i=0}^{k} p \cdot p_i\\
=\ &p \cdot \textstyle\sum_{i=0}^{k} p_i\\
=\ &p.
\end{aligned}$$

- $\varphi = t :: \varphi'$, $\Gamma = \mathbf{Tree}(p, W, \Gamma')$, $W \overset{t}{\rightsquigarrow} W'$, $\#\mathit{nextsplit}(W, t) > 1$ and $\mathbf{History}_T(p, W', \varphi', \Gamma')$.
  From $\mathbf{History}_T(p, W', \varphi', \Gamma')$ by IH we have $\sum_{s'} \mathit{prob}(\Gamma', s', n') = p$, thus
  $$\begin{aligned}
  &\textstyle\sum_s \mathit{prob}(\Gamma, s, n)\\
  =\ &\textstyle\sum_{i,s'}\{\mathit{prob}(\mathbf{Tree}(p, W, \Gamma'), i :: s', n'+1) \mid i \le 0\}\\
  =\ &\textstyle\sum_{s'} \mathit{prob}(\Gamma_i, s', n')\\
  =\ &p.
  \end{aligned}$$

Lemma 91. *For all $P, \mathbb{C}, Q$, if $\models_{AT} \{P\}\mathbb{C}\{Q\}$ and $\mathbf{closed}(Q)$, then $\models_{A'} \{P\}\mathbf{RemoveSplit}(\mathbb{C})\{Q\}$.*

Proof. For all $P, \mathbb{C}, Q$ such that $\models_{AT} \{P\}\mathbb{C}\{Q\}$ and $\mathbf{closed}(Q)$, by Def. 40 we need to prove for all μ, φ, μ', if $\mu \models P$ and $\mathit{init}(\mathbf{RemoveSplit}(\mathbb{C}), \mu) \Downarrow'_\varphi \mu'$, then $\mu' \models Q$. For all μ, φ, μ' such that $\mu \models P$ and $\mathit{init}(\mathbf{RemoveSplit}(\mathbb{C}), \mu) \Downarrow'_\varphi \mu'$, by Lem. 80 we know $\mathbf{RemoveSplit}(\mathit{init}(\mathbb{C}, \mu)) = \mathit{init}(\mathbf{RemoveSplit}(\mathbb{C}), \mu)$, so $\mathbf{RemoveSplit}(\mathit{init}(\mathbb{C}, \mu)) \Downarrow'_\varphi \mu'$. By Def. 39 there exists $\vec{W}$ such that $\mathbf{History}(\mathbf{RemoveSplit}(\mathit{init}(\mathbb{C}, \mu)), \varphi, \vec{W})$, $\lim_{n\to\infty} \vec{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$ and for all σ, $\lim_{n\to\infty} \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = \mu'(\sigma)$. By Lem. 81 there exists Γ such that $\mathbf{History}_T(1, W, \varphi, \Gamma)$. From $\mathbf{History}(\mathbf{RemoveSplit}(\mathit{init}(\mathbb{C}, \mu)), \varphi, \vec{W})$ by Lem. 88 we know for all σ and n, $\vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) =$
$\sum_{\mathbb{C}'} \delta(\mathbf{RemoveSplit}(\mathbb{C}'))(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \cdot \sum_s \mathit{prob}(\Gamma, s, n) \cdot \mathit{traverse}(\Gamma, s, n)(\mathbb{C}', \sigma)$
$= \sum_s \mathit{prob}(\Gamma, s, n) \cdot \mathit{traverse}(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$, i.e.,

$$\forall \sigma, n.\ \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = \sum_s \mathit{prob}(\Gamma, s, n) \cdot \mathit{traverse}(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma). \tag{49}$$

From $\mathbf{History}_T(1, W, \varphi, \Gamma)$ by Lem. 89 we know

$$\begin{aligned}\forall s, n, \sigma. &\mathit{prob}(\Gamma, s, n+1) \cdot \mathit{traverse}(\Gamma, s, n+1)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \ge\\ &\mathit{prob}(\Gamma, s, n) \cdot \mathit{traverse}(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma),\end{aligned} \tag{50}$$

thus

$$\begin{aligned}\forall s, n. &\textstyle\sum_\sigma \mathit{prob}(\Gamma, s, n+1) \cdot \mathit{traverse}(\Gamma, s, n+1)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \ge\\ &\textstyle\sum_\sigma \mathit{prob}(\Gamma, s, n) \cdot \mathit{traverse}(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma).\end{aligned}$$

By Monotone Convergence Theorem for Series we know

$$\forall s.\ \lim_{n\to\infty} \sum_s \sum_\sigma prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = \sum_s \lim_{n\to\infty} \sum_\sigma prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma). \tag{51}$$

Therefore

$$\begin{aligned}
1 &= \lim_{n\to\infty} \vec{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \\
&= \lim_{n\to\infty} \sum_\sigma \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \\
&= \lim_{n\to\infty} \sum_\sigma \sum_s prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) && \text{(by Eqn. (49))} \\
&= \lim_{n\to\infty} \sum_s \sum_\sigma prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) && \text{(by Tonelli's Theorem)} \\
&= \sum_s \lim_{n\to\infty} \sum_\sigma prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) && \text{(by Eqn. (51))} \\
&= \sum_s \lim_{n\to\infty} prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \\
&= \sum_s \lim_{n\to\infty} prob(\Gamma, s, n) \cdot \lim_{n\to\infty} traverse(\Gamma, s, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})
\end{aligned}$$

From $\mathbf{History}(1, W, \varphi, \Gamma)$ by Lem. 90 we know $\sum_s prob(\Gamma, s, n) = 1$ for all n. By Fatou's Lemma we know $\sum_s \lim_{n\to\infty} prob(\Gamma, s, n) \leq \lim_{n\to\infty} \sum_s prob(\Gamma, s, n) = \lim_{n\to\infty} 1 = 1$. Thus $\sum_s \lim_{n\to\infty} prob(\Gamma, s, n) \cdot \lim_{n\to\infty} traverse(\Gamma, s, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \geq \sum_s \lim_{n\to\infty} prob(\Gamma, s, n)$, so $\sum_s \lim_{n\to\infty} prob(\Gamma, s, n) \cdot (1 - \lim_{n\to\infty} traverse(\Gamma, s, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})) \leq 0$.
From $\forall s, n.\ traverse(\Gamma, s, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \leq 1$ we know $\lim_{n\to\infty} traverse(\Gamma, s, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \leq 1$ for all s, so $1 - \lim_{n\to\infty} traverse(\Gamma, s, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \geq 0$ for all s. From $\forall s, n.\ prob(\Gamma, s, n) \geq 0$ we know $\lim_{n\to\infty} prob(\Gamma, s, n) \geq 0$ for all s, so $\lim_{n\to\infty} prob(\Gamma, s, n) \cdot (1 - \lim_{n\to\infty} traverse(\Gamma, s, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})) \geq 0$ for all s. From $\sum_s \lim_{n\to\infty} prob(\Gamma, s, n) \cdot (1 - \lim_{n\to\infty} traverse(\Gamma, s, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})) \leq 0$ we know for all s, $\lim_{n\to\infty} prob(\Gamma, s, n) \cdot (1 - \lim_{n\to\infty} traverse(\Gamma, s, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})) = 0$. Thus

$$\forall s.\ \lim_{n\to\infty} prob(\Gamma, s, n) = 0 \vee \lim_{n\to\infty} traverse(\Gamma, s, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1. \tag{52}$$

From Eqn. (50) by Monotone Convergence Theorem for Series we know

$$\forall \sigma.\ \lim_{n\to\infty} \sum_s prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = \sum_s \lim_{n\to\infty} prob(\Gamma, s, n) \cdot traverse(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma). \tag{53}$$

Let $\mu'' \stackrel{\text{def}}{=} \lambda\nu \in \mathbb{D}_{State}. \sum_s \delta(\lambda\sigma. \lim_{n\to\infty} \mathit{traverse}(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma))(\nu) \cdot \lim_{n\to\infty} \mathit{prob}(\Gamma, s, n)$, then

$$
\begin{aligned}
\overline{\mu''} &= \lambda\sigma. \textstyle\sum_\nu \mu''(\nu) \cdot \nu(\sigma) \\
&= \lambda\sigma. \textstyle\sum_\nu \sum_s \delta(\lambda\sigma. \lim_{n\to\infty} \mathit{traverse}(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma))(\nu) \cdot \lim_{n\to\infty} \mathit{prob}(\Gamma, s, n) \cdot \nu(\sigma) \\
&= \lambda\sigma. \textstyle\sum_s \lim_{n\to\infty} \mathit{prob}(\Gamma, s, n) \cdot \lim_{n\to\infty} \mathit{traverse}(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \\
&= \lambda\sigma. \textstyle\sum_s \lim_{n\to\infty} \mathit{prob}(\Gamma, s, n) \cdot \mathit{traverse}(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \\
&= \lambda\sigma. \lim_{n\to\infty} \textstyle\sum_s \mathit{prob}(\Gamma, s, n) \cdot \mathit{traverse}(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \qquad \text{(by Eqn. (53))} \\
&= \lambda\sigma. \lim_{n\to\infty} \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \qquad \text{(by Eqn. (49))} \\
&= \mu'.
\end{aligned}
$$

From $\overline{\mu''} = \mu'$ and $\mathbf{closed}(Q)$, to prove $\mu' \models Q$, it suffices to prove for all $\nu \in \mathit{supp}(\mu'')$, $\nu \models Q$. For all $\nu \in \mathit{supp}(\mu'')$, we have $\mu''(\nu) > 0$, i.e., $\sum_s \delta(\lambda\sigma. \lim_{n\to\infty} \mathit{traverse}(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma))(\nu) \cdot \lim_{n\to\infty} \mathit{prob}(\Gamma, s, n) > 0$, so there exists s such that $\nu = \lambda\sigma. \lim_{n\to\infty} \mathit{traverse}(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$ and $\lim_{n\to\infty} \mathit{prob}(\Gamma, s, n) > 0$. From Eqn. (52) we know $\lim_{n\to\infty} \mathit{traverse}(\Gamma, s, n)^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$. From $\mathbf{History}_T(1, \mathit{init}(\mathbb{C}, \mu), \varphi, \Gamma)$ and $\nu = \lambda\sigma. \lim_{n\to\infty} \mathit{traverse}(\Gamma, s, n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma)$ we know $\mathit{init}(\mathbb{C}, \mu) \Downarrow^s_\varphi \nu$. From $\models_{AT} \{P\}\mathbb{C}\{Q\}$ and $\mu \models P$ we have $\nu \models Q$.

Lemma 92 (Soundness of (REMOVESPLIT) Rule). *For all $P, \mathbb{C}, Q$, if $\models_{\text{A}} \{P\}\mathbb{C}\{Q\}$ and $\mathbf{closed}(Q)$, then $\models_{\text{A}} \{P\}\mathbf{RemoveSplit}(\mathbb{C})\{Q\}$.*

Proof. For all $P, \mathbb{C}, Q$ such that $\models_{\text{A}} \{P\}\mathbb{C}\{Q\}$ and $\mathbf{closed}(Q)$, by Lem. 44 we know $\models_{\text{A}'} \{P\}\mathbb{C}\{Q\}$. By Lem. 78 we know $\models_{AT} \{P\}\mathbb{C}\{Q\}$. From $\mathbf{closed}(Q)$ by Lem. 91 we have $\models_{\text{A}'} \{P\}\mathbf{RemoveSplit}(\mathbb{C})\{Q\}$. By Lem. 44 we know $\models_{\text{A}} \{P\}\mathbf{RemoveSplit}(\mathbb{C})\{Q\}$.

$\mathbf{lazycoin}(\mathbf{skip})$	$\stackrel{\text{def}}{=} \mathbf{skip}$
$\mathbf{lazycoin}(x := e)$	$\stackrel{\text{def}}{=} x := e$
$\mathbf{lazycoin}(C_1; C_2)$	$\stackrel{\text{def}}{=} \mathbf{lazycoin}(C_1); \mathbf{lazycoin}(C_2)$
$\mathbf{lazycoin}(\mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2)$	$\stackrel{\text{def}}{=} \mathbf{if}\ (b)\ \mathbf{then}\ \mathbf{lazycoin}(C_1)\ \mathbf{else}\ \mathbf{lazycoin}(C_2)$
$\mathbf{lazycoin}(\mathbf{while}\ (b)\ \mathbf{do}\ C)$	$\stackrel{\text{def}}{=} \mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{lazycoin}(C)$
$\mathbf{lazycoin}(\langle C \rangle)$	$\stackrel{\text{def}}{=} \langle C \rangle$
$\mathbf{lazycoin}(\langle C \rangle\ sp)$	$\stackrel{\text{def}}{=} \langle C \rangle\ sp$
$\mathbf{lazycoin}(\langle C_1 \rangle \oplus_p \langle C_2 \rangle)$	$\stackrel{\text{def}}{=} \mathbf{skip};\ \langle \langle C_1 \rangle \oplus_p \langle C_2 \rangle \rangle$

Fig. 34: Definition of $\mathbf{lazycoin}(C)$

Definition 54. $\mathbf{lazycoin}(C_1 \parallel \cdots \parallel C_n) \stackrel{\text{def}}{=} \mathbf{lazycoin}(C_1) \parallel \cdots \parallel \mathbf{lazycoin}(C_n)$.
The definition of $\mathbf{lazycoin}(C)$ is given in Fig. 34.

Definition 55. $\mathbf{lazycoin}(W) \stackrel{\text{def}}{=} \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\delta(\mathbf{lazycoin}(\mathbb{C})) \otimes \delta(\sigma)\}$.

Definition 56. $\mathbf{lazycoin}(\eta) \stackrel{\text{def}}{=} \mathbb{E}_{(C,\sigma)\sim\eta}\{\delta(\mathbf{lazycoin}(C)) \otimes \delta(\sigma)\}$.

Definition 57. $\rho \in \mathbb{D}_{Stmt}$.

Definition 58. $\mathbf{lazycoin}(\rho) \stackrel{\text{def}}{=} \mathbb{E}_{C\sim\rho}\{\delta(\mathbf{lazycoin}(C))\}$.

Definition 59. $\rho_1 \parallel \cdots \parallel \rho_n \stackrel{\text{def}}{=} \lambda(C_1 \parallel \cdots \parallel C_n).\ \rho_1(C_1) \cdots \rho_n(C_n)$.

Definition 60. $\eta; C_2 \stackrel{\text{def}}{=} \lambda(C,\sigma).\ \begin{cases} \eta(C_1,\sigma), & \text{if } C = C_1; C_2 \\ 0, & \text{otherwise} \end{cases}$.

Definition 61. $\rho; C_2 \stackrel{\text{def}}{=} \lambda C.\ \begin{cases} \rho(C_1), & \text{if } C = C_1; C_2 \\ 0, & \text{otherwise} \end{cases}$.

Definition 62. $\mathbf{splitAtom}(C) \stackrel{\text{def}}{=} \begin{cases} \delta(\langle C_1\rangle) \oplus_p \delta(\langle C_2\rangle), & \text{if } C = \langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle \\ \delta(\langle C_1\rangle\ sp) \oplus_p \delta(\langle C_2\rangle\ sp), & \text{if } C = \langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle\ sp \\ \mathbf{splitAtom}(C_1); C_2, & \text{if } C = C_1; C_2 \\ \delta(C), & \text{otherwise} \end{cases}$

Definition 63. $\mathbf{splitAtom}(C_1 \parallel \cdots \parallel C_n) \stackrel{\text{def}}{=} \mathbf{splitAtom}(C_1) \parallel \cdots \parallel \mathbf{splitAtom}(C_n)$.

Definition 64. $\mathbf{splitAtom}(W) \stackrel{\text{def}}{=} \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{splitAtom}(\mathbb{C}) \otimes \delta(\sigma)\}$.

Definition 65. $\mathbf{splitAtom}(\eta) \stackrel{\text{def}}{=} \mathbb{E}_{(C,\sigma)\sim\eta}\{\mathbf{splitAtom}(C) \otimes \delta(\sigma)\}$.

Definition 66. $\mathbf{splitAtom}(\rho) \stackrel{\text{def}}{=} \mathbb{E}_{C\sim\rho}\{\mathbf{splitAtom}(C)\}$.

Definition 67. $W_1 \sim W_2$ if and only if $\mathbf{lazycoin}(W_1) = \mathbf{splitAtom}(W_2)$.

Definition 68. $\rho_1 \parallel \cdots \parallel \rho_{t-1} \parallel \eta \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n \stackrel{\text{def}}{=} \lambda(C_1 \parallel \cdots \parallel C_n, \sigma).\ \rho_1(C_1) \cdots \rho_{t-1}(C_{t-1}) \cdot \eta(C_t,\sigma) \cdot \rho_{t+1}(C_{t+1}) \cdots \rho_n(C_n)$.

Definition 69. $\mathbf{step}(C,\sigma) \stackrel{\text{def}}{=} \lambda(C',\sigma').\ \begin{cases} p, & \text{if } (C,\sigma) \xrightarrow{p} (C',\sigma') \\ 0, & \text{otherwise} \end{cases}$

Definition 70. $\mathbf{step}(C_1 \parallel \cdots \parallel C_n, \sigma, t) \stackrel{\text{def}}{=} \delta(C_1) \parallel \ldots \delta(C_{t-1}) \parallel \mathbf{step}(C_t,\sigma) \parallel \delta(C_{t+1}) \parallel \cdots \parallel \delta(C_n)$.

Definition 71. $\mathbf{step}(W,t) \stackrel{\text{def}}{=} \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{step}(\mathbb{C},\sigma,t)\}$.

Definition 72. $\mathbf{step}(\eta) \stackrel{\text{def}}{=} \mathbb{E}_{(C,\sigma)\sim\eta}\{\mathbf{step}(C,\sigma)\}$.

Lemma 93. *For all* ρ, μ, C_2, $(\rho \otimes \mu); C_2 = (\rho; C_2) \otimes \mu$.

Proof. For all ρ, μ, C_2,

$$\begin{aligned}
&(\rho \otimes \mu); C_2 \\
&= \lambda(C,\sigma).\begin{cases}(\rho\otimes\mu)(C_1,\sigma), & \text{if } C = C_1; C_2\\ 0, & \text{otherwise}\end{cases}\\
&= \lambda(C,\sigma).\begin{cases}\rho(C_1)\cdot\mu(\sigma), & \text{if } C = C_1; C_2\\ 0, & \text{otherwise}\end{cases}\\
&= \lambda(C,\sigma).\ \mu(\sigma)\cdot\begin{cases}\rho(C_1), & \text{if } C = C_1; C_2\\ 0, & \text{otherwise}\end{cases}\\
&= \lambda(C,\sigma).\ \mu(\sigma)\cdot(\rho; C_2)(C)\\
&= (\rho; C_2)\otimes\mu.
\end{aligned}$$

Lemma 94. *For all* $\mathbb{C}$ *and* μ, $\mathbf{lazycoin}(\delta(\mathbb{C})\otimes\mu) = \delta(\mathbf{lazycoin}(\mathbb{C}))\otimes\mu$.

Proof. For all $\mathbb{C}$ and μ, we have

$$\begin{aligned}
&\mathbf{lazycoin}(\delta(\mathbb{C})\otimes\mu)\\
&= \mathbb{E}_{(\mathbb{C}_1,\sigma)\sim\delta(\mathbb{C})\otimes\mu}\{\delta(\mathbf{lazycoin}(\mathbb{C}_1))\otimes\delta(\sigma)\}\\
&= \lambda(\mathbb{C}',\sigma').\ \textstyle\sum_{\mathbb{C}_1,\sigma}\delta(\mathbb{C})(\mathbb{C}_1)\cdot\mu(\sigma)\cdot\delta(\mathbf{lazycoin}(\mathbb{C}_1))(\mathbb{C}')\cdot\delta(\sigma)(\sigma')\\
&= \lambda(\mathbb{C}',\sigma').\ \delta(\mathbf{lazycoin}(\mathbb{C}))(\mathbb{C}')\cdot\mu(\sigma')\\
&= \delta(\mathbf{lazycoin}(\mathbb{C}))\otimes\mu.
\end{aligned}$$

Lemma 95. *For all* $\mathbb{C}$ *and* μ, $\mathbf{splitAtom}(\delta(\mathbb{C})\otimes\mu) = \mathbf{splitAtom}(\mathbb{C})\otimes\mu$.

Proof. For all $\mathbb{C}$ and μ, we have

$$\begin{aligned}
&\mathbf{splitAtom}(\delta(\mathbb{C})\otimes\mu)\\
&= \mathbb{E}_{(\mathbb{C}_1,\sigma)\sim\delta(\mathbb{C})\otimes\mu}\{\mathbf{splitAtom}(\mathbb{C}_1)\otimes\delta(\sigma)\}\\
&= \lambda(\mathbb{C}',\sigma').\ \textstyle\sum_{\mathbb{C}_1,\sigma}\delta(\mathbb{C})(\mathbb{C}_1)\cdot\mu(\sigma)\cdot\mathbf{splitAtom}(\mathbb{C}_1)(\mathbb{C}')\cdot\delta(\sigma)(\sigma')\\
&= \lambda(\mathbb{C}',\sigma').\ \mathbf{splitAtom}(\mathbb{C})(\mathbb{C}')\cdot\mu(\sigma')\\
&= \mathbf{splitAtom}(\mathbb{C})\otimes\mu.
\end{aligned}$$

Lemma 96. *For all* C_1 *and* C_2, $\delta(C_1); C_2 = \delta(C_1; C_2)$.

Proof. For all C_1 and C_2,

$$\delta(C_1); C_2 = \lambda C'.\begin{cases}\delta(C_1)(C_1'), & \text{if } C' = C_1'; C_2\\ 0, & \text{otherwise}\end{cases} = \lambda C'.\begin{cases}1, & \text{if } C' = C_1; C_2\\ 0, & \text{otherwise}\end{cases} = \delta(C_1; C_2).$$

Lemma 97. *For all* C, $\mathbf{splitAtom}(\mathbf{lazycoin}(C)) = \delta(\mathbf{lazycoin}(C))$.

Proof. by induction on C.

- $C = \mathbf{skip}$.
  $\mathbf{splitAtom}(\mathbf{lazycoin}(C)) = \mathbf{splitAtom}(\mathbf{lazycoin}(\mathbf{skip})) = \mathbf{splitAtom}(\mathbf{skip}) = \delta(\mathbf{skip}) = \delta(\mathbf{lazycoin}(\mathbf{skip})) = \delta(\mathbf{lazycoin}(\mathbb{C}))$.
- $C = x := e$.
  $\mathbf{splitAtom}(\mathbf{lazycoin}(C)) = \mathbf{splitAtom}(\mathbf{lazycoin}(x := e)) = \mathbf{splitAtom}(x := e) = \delta(x := e) = \delta(\mathbf{lazycoin}(x := e)) = \delta(\mathbf{lazycoin}(\mathbb{C}))$.
- $C = C_1; C_2$.
  IH: $\mathbf{splitAtom}(\mathbf{lazycoin}(C_1)) = \delta(\mathbf{lazycoin}(C_1))$.

$$
\begin{aligned}
&\mathbf{splitAtom}(\mathbf{lazycoin}(C)) \\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(C_1; C_2)) \\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(C_1); \mathbf{lazycoin}(C_2)) \\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(C_1)); \mathbf{lazycoin}(C_2) \\
=\ &\delta(\mathbf{lazycoin}(C_1)); \mathbf{lazycoin}(C_2) && \text{(by IH)} \\
=\ &\delta(\mathbf{lazycoin}(C_1); \mathbf{lazycoin}(C_2)) && \text{(by Lem. 96)} \\
=\ &\delta(\mathbf{lazycoin}(C_1; C_2)) \\
=\ &\delta(\mathbf{lazycoin}(C)).
\end{aligned}
$$

- $C = \mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2$.

$$
\begin{aligned}
&\mathbf{splitAtom}(\mathbf{lazycoin}(C)) \\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(\mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2)) \\
=\ &\mathbf{splitAtom}(\mathbf{if}\ (b)\ \mathbf{then}\ \mathbf{lazycoin}(C_1)\ \mathbf{else}\ \mathbf{lazycoin}(C_2)) \\
=\ &\delta(\mathbf{if}\ (b)\ \mathbf{then}\ \mathbf{lazycoin}(C_1)\ \mathbf{else}\ \mathbf{lazycoin}(C_2)) \\
=\ &\delta(\mathbf{lazycoin}(\mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2)) \\
=\ &\delta(\mathbf{lazycoin}(C)).
\end{aligned}
$$

- $C = \mathbf{while}\ (b)\ \mathbf{do}\ C_1$.

$$
\begin{aligned}
&\mathbf{splitAtom}(\mathbf{lazycoin}(C)) \\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(\mathbf{while}\ (b)\ \mathbf{do}\ C_1)) \\
=\ &\mathbf{splitAtom}(\mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{lazycoin}(C_1)) \\
=\ &\delta(\mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{lazycoin}(C_1)) \\
=\ &\delta(\mathbf{lazycoin}(\mathbf{while}\ (b)\ \mathbf{do}\ C_1)) \\
=\ &\delta(\mathbf{lazycoin}(C)).
\end{aligned}
$$

- $C = \langle C_1 \rangle$.

$$
\begin{aligned}
&\mathbf{splitAtom}(\mathbf{lazycoin}(C)) \\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(\langle C_1 \rangle)) \\
=\ &\mathbf{splitAtom}(\langle C_1 \rangle) \\
=\ &\delta(\langle C_1 \rangle) \\
=\ &\delta(\mathbf{lazycoin}(\langle C_1 \rangle)) \\
=\ &\delta(\mathbf{lazycoin}(C)).
\end{aligned}
$$

– $C = \langle C_1 \rangle\ sp$.

$$\begin{aligned}
&\mathbf{splitAtom}(\mathbf{lazycoin}(C)) \\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(\langle C_1 \rangle\ sp)) \\
=\ &\mathbf{splitAtom}(\langle C_1 \rangle\ sp) \\
=\ &\delta(\langle C_1 \rangle\ sp) \\
=\ &\delta(\mathbf{lazycoin}(\langle C_1 \rangle\ sp)) \\
=\ &\delta(\mathbf{lazycoin}(C)).
\end{aligned}$$

– $C = \langle C_1 \rangle \oplus_p \langle C_2 \rangle$.

$$\begin{aligned}
&\mathbf{splitAtom}(\mathbf{lazycoin}(C)) \\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(\langle C_1 \rangle \oplus_p \langle C_2 \rangle)) \\
=\ &\mathbf{splitAtom}(\mathbf{skip}; \langle \langle C_1 \rangle \oplus_p \langle C_2 \rangle \rangle) \\
=\ &\mathbf{splitAtom}(\mathbf{skip}); \langle \langle C_1 \rangle \oplus_p \langle C_2 \rangle \rangle \\
=\ &\delta(\mathbf{skip}); \langle \langle C_1 \rangle \oplus_p \langle C_2 \rangle \rangle \\
=\ &\delta(\mathbf{skip}; \langle \langle C_1 \rangle \oplus_p \langle C_2 \rangle \rangle) \qquad \text{(by Lem. 96)} \\
=\ &\delta(\mathbf{lazycoin}(\langle C_1 \rangle \oplus_p \langle C_2 \rangle)) \\
=\ &\delta(\mathbf{lazycoin}(C)).
\end{aligned}$$

Lemma 98. *For all* $C_1, \ldots, C_n$, $\delta(C_1) \parallel \cdots \parallel \delta(C_n) = \delta(C_1 \parallel \cdots \parallel C_n)$..

Proof. For all $C_1, \ldots, C_n$,

$$\begin{aligned}
&\delta(C_1) \parallel \cdots \parallel \delta(C_n) \\
=\ &\lambda(C_1' \parallel \cdots \parallel C_n').\ \delta(C_1)(C_1') \cdots \delta(C_n)(C_n') \\
=\ &\lambda(C_1' \parallel \cdots \parallel C_n').\ \delta(C_1 \parallel \cdots \parallel C_n)(C_1' \parallel \cdots \parallel C_n') \\
=\ &\delta(C_1 \parallel \cdots \parallel C_n).
\end{aligned}$$

Lemma 99. *For all* $\mathbb{C}$, $\mathbf{splitAtom}(\mathbf{lazycoin}(\mathbb{C})) = \delta(\mathbf{lazycoin}(\mathbb{C}))$.

Proof. For all $\mathbb{C}$, by definition of *Prog* there exists $C_1, \ldots, C_n$ such that $\mathbb{C} = C_1 \parallel \cdots \parallel C_n$.

$$\begin{aligned}
&\mathbf{splitAtom}(\mathbf{lazycoin}(\mathbb{C})) \\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(C_1 \parallel \cdots \parallel C_n)) \\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(C_1) \parallel \cdots \parallel \mathbf{lazycoin}(C_n)) \\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(C_1)) \parallel \cdots \parallel \mathbf{splitAtom}(\mathbf{lazycoin}(C_n)) \\
=\ &\delta(\mathbf{lazycoin}(C_1)) \parallel \cdots \parallel \delta(\mathbf{lazycoin}(C_n)) \qquad \text{(by Lem. 97)} \\
=\ &\delta(\mathbf{lazycoin}(C_1) \parallel \cdots \parallel \mathbf{lazycoin}(C_n)) \qquad \text{(by Lem. 98)} \\
=\ &\delta(\mathbf{lazycoin}(C_1 \parallel \cdots \parallel C_n)) \\
=\ &\delta(\mathbf{lazycoin}(\mathbb{C})).
\end{aligned}$$

Lemma 100. *For all* $\mathbb{C}$ *and* μ, $\mathit{init}(\mathbb{C}, \mu) \sim \mathit{init}(\mathbf{lazycoin}(\mathbb{C}), \mu)$.

Proof. For all $\mathbb{C}$ and μ, we have

$$\begin{aligned}
&\mathbf{splitAtom}(\mathit{init}(\mathbf{lazycoin}(\mathbb{C}), \mu)) \\
=\ &\mathbf{splitAtom}(\delta(\mathbf{lazycoin}(\mathbb{C})) \otimes \mu) \\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(\mathbb{C})) \otimes \mu \qquad \text{(by Lem. 95)} \\
=\ &\delta(\mathbf{lazycoin}(\mathbb{C})) \otimes \mu \qquad \text{(by Lem. 99)} \\
=\ &\mathbf{lazycoin}(\delta(\mathbb{C}) \otimes \mu) \qquad \text{(by Lem. 94)} \\
=\ &\mathbf{lazycoin}(\mathit{init}(\mathbb{C}, \mu)).
\end{aligned}$$

Therefore $\mathit{init}(\mathbb{C}, \mu) \sim \mathit{init}(\mathbf{lazycoin}(\mathbb{C}), \mu)$.

Lemma 101. *For all C, $\mathit{nextsplit}(\mathbf{lazycoin}(C)) = \mathit{nextsplit}(C)$.*

Proof. by induction on the structure of C.

- case 1: $C = \langle C_1 \rangle\ sp$.
  $\mathit{nextsplit}(\mathbf{lazycoin}(C)) = \mathit{nextsplit}(\mathbf{lazycoin}(\langle C_1 \rangle\ sp)) = \mathit{nextsplit}(\langle C \rangle_1\ sp) = \mathit{nextsplit}(C)$.
- case 2: $C = C_1; C_2$.
  IH: $\mathit{nextsplit}(\mathbf{lazycoin}(C_1)) = \mathit{nextsplit}(C_1)$.
  $\mathit{nextsplit}(\mathbf{lazycoin}(C)) = \mathit{nextsplit}(\mathbf{lazycoin}(C_1; C_2)) = \mathit{nextsplit}(\mathbf{lazycoin}(C_1); \mathbf{lazycoin}(C_2))$
  $= \mathit{nextsplit}(\mathbf{lazycoin}(C_1)) = \mathit{nextsplit}(C_1)$.
- othercases.
  $\mathit{nextsplit}(\mathbf{lazycoin}(C)) = \{\mathbf{split}(\mathrm{true})\} = \mathit{nextsplit}(C)$.

Lemma 102. *For all W and t, $\mathit{nextsplit}(\mathbf{lazycoin}(W), t) = \mathit{nextsplit}(W, t)$.*

Proof. For all W and t,

$$
\begin{aligned}
&\mathit{nextsplit}(\mathbf{lazycoin}(W), t) \\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma.\ (C_1 \parallel \cdots \parallel C_n, \sigma) \in \mathit{supp}(\mathbf{lazycoin}(W))\} \\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma.\ \mathbf{lazycoin}(W)(C_1 \parallel \cdots \parallel C_n, \sigma) > 0\} \\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma. \\
&\qquad \mathbb{E}_{(\mathbb{C}', \sigma') \sim W}\{\delta(\mathbf{lazycoin}(\mathbb{C}')) \otimes \delta(\sigma')\}(C_1 \parallel \cdots \parallel C_n, \sigma) > 0\} \\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma. \\
&\qquad \textstyle\sum_{\mathbb{C}', \sigma'} W(\mathbb{C}', \sigma') \cdot \delta(\mathbf{lazycoin}(\mathbb{C}'))(C_1 \parallel \cdots \parallel C_n) \cdot \delta(\sigma')(\sigma) > 0\} \\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma. \\
&\qquad \textstyle\sum_{\mathbb{C}'} W(\mathbb{C}', \sigma) \cdot \delta(\mathbf{lazycoin}(\mathbb{C}'))(C_1 \parallel \cdots \parallel C_n) > 0\} \\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma, \mathbb{C}'. \\
&\qquad W(\mathbb{C}', \sigma) > 0 \wedge \mathbf{lazycoin}(\mathbb{C}') = C_1 \parallel \cdots \parallel C_n\} \\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma, C'_1, \ldots, C'_k. \\
&\qquad W(C'_1 \parallel \cdots \parallel C'_k, \sigma) > 0 \wedge \mathbf{lazycoin}(C'_1 \parallel \cdots \parallel C'_k) = C_1 \parallel \cdots \parallel C_n\} \\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma, C'_1, \ldots, C'_k.\ W(C'_1 \parallel \cdots \parallel C'_k, \sigma) > 0 \wedge \\
&\qquad \mathbf{lazycoin}(C'_1) \parallel \cdots \parallel \mathbf{lazycoin}(C'_k) = C_1 \parallel \cdots \parallel C_n\} \\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma, C'_1, \ldots, C'_n.\ W(C'_1 \parallel \cdots \parallel C'_n, \sigma) > 0 \wedge \\
&\qquad \mathbf{lazycoin}(C'_1) = C_1 \wedge \cdots \wedge \mathbf{lazycoin}(C'_n) = C_n\} \\
&= \{\mathit{nextsplit}(C_t) \mid \exists C'_1, \ldots, C'_n, \sigma.\ W(C'_1 \parallel \cdots \parallel C'_n, \sigma) > 0 \wedge \mathbf{lazycoin}(C'_t) = C_t\} \\
&= \{\mathit{nextsplit}(\mathbf{lazycoin}(C'_t)) \mid \exists C'_1, \ldots, C'_{t-1}, C'_{t+1}, \ldots, C'_n, \sigma.\ W(C'_1 \parallel \cdots \parallel C'_n, \sigma) > 0\} \\
&= \{\mathit{nextsplit}(C'_t) \mid \exists C'_1, \ldots, C'_{t-1}, C'_{t+1}, \ldots, C'_n, \sigma.\ W(C'_1 \parallel \cdots \parallel C'_n, \sigma) > 0\} \quad \text{(by Lem. 101)} \\
&= \{\mathit{nextsplit}(C'_t) \mid \exists C'_1, \ldots, C'_{t-1}, C'_{t+1}, \ldots, C'_n, \sigma.\ (C'_1 \parallel \cdots \parallel C'_n, \sigma) \in \mathit{supp}(W)\} \\
&= \mathit{nextsplit}(W, t).
\end{aligned}
$$

Lemma 103. *For all C and C', if $\mathbf{splitAtom}(C)(C') > 0$, then $\mathit{nextsplit}(C) = \mathit{nextsplit}(C')$.*

Proof. by induction on the structure of C.

– case 1: $C = \langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle$.
For all C' such that $\mathbf{splitAtom}(C)(C') > 0$, we have $0 < \mathbf{splitAtom}(C)(C') =$

$\mathbf{splitAtom}(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle)(C') = (\delta(\langle C_1\rangle) \oplus_p \delta(\langle C_2\rangle))(C') = p \cdot \delta(\langle C_1\rangle)(C') + (1-p) \cdot \delta(\langle C_2\rangle)(C')$ we know $\delta(\langle C_1\rangle)(C') > 0$ or $\delta(\langle C_2\rangle)(C') > 0$, so $C' = \langle C_1\rangle$ or $C' = \langle C_2\rangle$, thus $\mathit{nextsplit}(C') = \{\mathbf{split}(\text{true})\} = \mathit{nextsplit}(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle) = \mathit{nextsplit}(C)$.

– case 2: $C = \langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle\ sp$.
For all C' such that $\mathbf{splitAtom}(C)(C') > 0$, we have $0 < \mathbf{splitAtom}(C)(C') =$

$\mathbf{splitAtom}(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle\ sp)(C') = (\delta(\langle C_1\rangle\ sp) \oplus_p \delta(\langle C_2\rangle)\ sp)(C') = p \cdot \delta(\langle C_1\rangle\ sp)(C') + (1-p) \cdot \delta(\langle C_2\rangle)(C'\ sp)$ we know $\delta(\langle C_1\rangle\ sp)(C') > 0$ or $\delta(\langle C_2\rangle\ sp)(C') > 0$, so $C' = \langle C_1\rangle\ sp$ or $C' = \langle C_2\rangle\ sp$, thus $\mathit{nextsplit}(C') = \{sp\} = \mathit{nextsplit}(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle\ sp) = \mathit{nextsplit}(C)$.

– case 3: $C = C_1; C_2$.
IH: for all C', if $\mathbf{splitAtom}(C_1)(C')$, then $\mathit{nextsplit}(C_1) = \mathit{nextsplit}(C')$.
From $0 < \mathbf{splitAtom}(C)(C') = \mathbf{splitAtom}(C_1; C_2)(C') = (\mathbf{splitAtom}(C_1); C_2)(C')$ we know there exists C_1' such that $C' = C_1'; C_2$ and $\mathbf{splitAtom}(C_1)(C_1') > 0$,
by IH we have $\mathit{nextsplit}(C_1) = \mathit{nextsplit}(C_1')$, thus $\mathit{nextsplit}(C) = \mathit{nextsplit}(C_1; C_2) =$
$\mathit{nextsplit}(C_1) = \mathit{nextsplit}(C_1') =$
$\mathit{nextsplit}(C_1'; C_2) = \mathit{nextsplit}(C')$.

– other cases.
By definition of **splitAtom** we know $\mathbf{splitAtom}(C) = \delta(C)$. For all C' such that
$\mathbf{splitAtom}(C)(C') > 0$, we have $\delta(C)(C') > 0$, so $C = C'$, thus $\mathit{nextsplit}(C) = \mathit{nextsplit}(C')$.

Lemma 104. *For all W and t, $\mathit{nextsplit}(\mathbf{splitAtom}(W), t) = \mathit{nextsplit}(W, t)$.*

Proof. For all W and t,

$$
\begin{aligned}
&\mathit{nextsplit}(\mathbf{splitAtom}(W), t)\\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma.\ (C_1 \parallel \cdots \parallel C_n, \sigma) \in \mathit{supp}(\mathbf{splitAtom}(W))\}\\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma.\ \mathbf{splitAtom}(W)(C_1 \parallel \cdots \parallel C_n, \sigma) > 0\}\\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma.\\
&\qquad \mathbb{E}_{(\mathbb{C}', \sigma') \sim W}\{\mathbf{splitAtom}(\mathbb{C}') \otimes \delta(\sigma')\}(C_1 \parallel \cdots \parallel C_n, \sigma) > 0\}\\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma.\\
&\qquad \textstyle\sum_{\mathbb{C}', \sigma'} W(\mathbb{C}', \sigma') \cdot \mathbf{splitAtom}(\mathbb{C}')(C_1 \parallel \cdots \parallel C_n) \cdot \delta(\sigma')(\sigma) > 0\}\\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma.\\
&\qquad \textstyle\sum_{\mathbb{C}'} W(\mathbb{C}', \sigma) \cdot \mathbf{splitAtom}(\mathbb{C}')(C_1 \parallel \cdots \parallel C_n) > 0\}\\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma, \mathbb{C}'.\\
&\qquad W(\mathbb{C}', \sigma) > 0 \wedge \mathbf{splitAtom}(\mathbb{C}')(C_1 \parallel \cdots \parallel C_n) > 0\}\\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma, C'_1, \ldots, C'_k.\\
&\qquad W(C'_1 \parallel \cdots \parallel C'_k, \sigma) > 0 \wedge \mathbf{splitAtom}(C'_1 \parallel \cdots \parallel C'_k)(C_1 \parallel \cdots \parallel C_n) > 0\}\\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma, C'_1, \ldots, C'_n.\ W(C'_1 \parallel \cdots \parallel C'_k, \sigma) > 0 \wedge\\
&\qquad \textstyle\prod_{i=1}^{n} \mathbf{splitAtom}(C'_i)(C_i) > 0\}\\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma, C'_1, \ldots, C'_n.\ W(C'_1 \parallel \cdots \parallel C'_n, \sigma) > 0 \wedge\\
&\qquad \mathbf{splitAtom}(C'_1)(C_1) > 0 \wedge \cdots \wedge \mathbf{splitAtom}(C'_n)(C_n) > 0\}\\
&= \{\mathit{nextsplit}(C_t) \mid \exists C'_1, \ldots, C'_n, \sigma.\ W(C'_1 \parallel \cdots \parallel C'_n, \sigma) > 0 \wedge \mathbf{splitAtom}(C'_t)(C_t) > 0\}\\
&= \{\mathit{nextsplit}(C_t) \mid \exists C'_1, \ldots, C'_{t-1}, C'_{t+1}, \ldots, C'_n, \sigma.\ W(C'_1 \parallel \cdots \parallel C'_n, \sigma) > 0 \wedge\\
&\qquad \mathbf{splitAtom}(C'_t)(C_t) > 0 \wedge \mathit{nextsplit}(C_t) = \mathit{nextsplit}(C'_t)\} \quad \text{(by Lem. 103)}\\
&= \{\mathit{nextsplit}(C'_t) \mid \exists C'_1, \ldots, C'_{t-1}, C'_{t+1}, \ldots, C'_n, \sigma, C_t.\ W(C'_1 \parallel \cdots \parallel C'_n, \sigma) > 0 \wedge\\
&\qquad \mathbf{splitAtom}(C'_t)(C_t) > 0 \wedge \mathit{nextsplit}(C_t) = \mathit{nextsplit}(C'_t)\}\\
&= \{\mathit{nextsplit}(C'_t) \mid \exists C'_1, \ldots, C'_{t-1}, C'_{t+1}, \ldots, C'_n, \sigma.\ W(C'_1 \parallel \cdots \parallel C'_n, \sigma) > 0\}\\
&= \{\mathit{nextsplit}(C'_t) \mid \exists C'_1, \ldots, C'_{t-1}, C'_{t+1}, \ldots, C'_n, \sigma.\ (C'_1 \parallel \cdots \parallel C'_n, \sigma) \in \mathit{supp}(W)\}\\
&= \mathit{nextsplit}(W, t).
\end{aligned}
$$

Lemma 105. *For all W_1, W_2, t, if $W_1 \sim W_2$, then $\mathit{nextsplit}(W_1, t) = \mathit{nextsplit}(W_2, t)$.*

Proof. For all W_1, W_2, t, if $W_1 \sim W_2$, we know $\mathbf{lazycoin}(W_1) = \mathbf{splitAtom}(W_2)$.
By Lem. 102 we know $\mathit{nextsplit}(\mathbf{lazycoin}(W), t) = \mathit{nextsplit}(W, t)$.
By Lem. 104 we know $\mathit{nextsplit}(\mathbf{splitAtom}(W), t) = \mathit{nextsplit}(W, t)$.
Thus $\mathit{nextsplit}(W_1, t) = \mathit{nextsplit}(\mathbf{lazycoin}(W), t) = \mathit{nextsplit}(\mathbf{splitAtom}(W), t) = \mathit{nextsplit}(W_2, t)$.

Lemma 106. *For all W, $\mathbf{lazycoin}(W)^{(State)} = W^{(State)}$.*

Proof. For all W,

$$
\begin{aligned}
&\mathbf{lazycoin}(W)^{(State)}\\
&= \lambda\sigma. \textstyle\sum_{\mathbb{C}} \mathbf{lazycoin}(W)(\mathbb{C}, \sigma)\\
&= \lambda\sigma. \textstyle\sum_{\mathbb{C}} \mathbb{E}_{(\mathbb{C}_1, \sigma_1) \sim W}\{\delta(\mathbf{lazycoin}(\mathbb{C}_1)) \otimes \delta(\sigma_1)\}(\mathbb{C}, \sigma)\\
&= \lambda\sigma. \textstyle\sum_{\mathbb{C}} \sum_{\mathbb{C}_1, \sigma_1} W(\mathbb{C}_1, \sigma_1) \cdot \delta(\mathbf{lazycoin}(\mathbb{C}_1))(\mathbb{C}) \cdot \delta(\sigma_1)(\sigma)\\
&= \lambda\sigma. \textstyle\sum_{\mathbb{C}_1} W(\mathbb{C}_1, \sigma)\\
&= W^{(State)}.
\end{aligned}
$$

Lemma 107. *For all W, $\mathbf{splitAtom}(W)^{(State)} = W^{(State)}$.*

Proof. For all W,

$$\begin{aligned}
&\mathbf{splitAtom}(W)^{(State)}\\
&= \lambda\sigma.\sum_{\mathbb{C}} \mathbf{splitAtom}(W)(\mathbb{C},\sigma)\\
&= \lambda\sigma.\sum_{\mathbb{C}} \mathbb{E}_{(\mathbb{C}_1,\sigma_1)\sim W}\{\mathbf{splitAtom}(\mathbb{C}_1)\otimes\delta(\sigma_1)\}(\mathbb{C},\sigma)\\
&= \lambda\sigma.\sum_{\mathbb{C}}\sum_{\mathbb{C}_1,\sigma_1} W(\mathbb{C}_1,\sigma_1)\cdot\mathbf{splitAtom}(\mathbb{C}_1)(\mathbb{C})\cdot\delta(\sigma_1)(\sigma)\\
&= \lambda\sigma.\sum_{\mathbb{C}_1} W(\mathbb{C}_1,\sigma)\\
&= W^{(State)}.
\end{aligned}$$

Lemma 108. *For all W_1, W_2, if $W_1 \sim W_2$, then ${W_1}^{(State)} = {W_2}^{(State)}$.*

Proof. For all W_1, W_2 such that $W_1 \sim W_2$, we know $\mathbf{lazycoin}(W_1) = \mathbf{splitAtom}(W_2)$. By Lem. 106 we know $\mathbf{lazycoin}(W_1)^{(State)} = {W_1}^{(State)}$. By Lem. 107 we know $\mathbf{splitAtom}(W_2)^{(State)} = {W_2}^{(State)}$. Therefore ${W_1}^{(State)} = \mathbf{lazycoin}(W_1)^{(State)} = \mathbf{splitAtom}(W_2)^{(State)} = {W_2}^{(State)}$.

Lemma 109. *For all W, t, W', if $W \overset{t}{\rightsquigarrow} W'$, then $W' = \boldsymbol{step}(W,t)$.*

Proof. For all W, t, W' such that $W \overset{t}{\rightsquigarrow} W'$, we have

$$\begin{aligned}
W' &= \lambda(\mathbb{C}',\sigma').\sum_{\mathbb{C},\sigma}\{W(\mathbb{C},\sigma)\cdot p \mid (\mathbb{C},\sigma)\xrightarrow[t]{p}(\mathbb{C}',\sigma')\}\\
&= \lambda(C_1' \parallel \ldots C_n',\sigma').\sum_{C_1,\ldots,C_n,\sigma}\{W(C_1\parallel\cdots\parallel C_n,\sigma)\cdot p \mid\\
&\qquad (C_1\parallel\cdots\parallel C_n,\sigma)\xrightarrow[t]{p}(C_1'\parallel\cdots\parallel C_n',\sigma')\}\\
&= \lambda(C_1' \parallel \ldots C_n',\sigma').\sum_{C_1,\ldots,C_n,\sigma}\{W(C_1\parallel\cdots\parallel C_n,\sigma)\cdot p \mid (C_t,\sigma)\xrightarrow{p}(C_t',\sigma')\wedge\\
&\qquad C_1'=C_1\wedge\cdots\wedge C_{t-1}'=C_{t-1}\wedge C_{t+1}'=C_{t+1}\wedge\cdots\wedge C_n'=C_n\}\\
&= \lambda(C_1' \parallel \ldots C_n',\sigma').\sum_{C_1,\ldots,C_n,\sigma} W(C_1\parallel\cdots\parallel C_n,\sigma)\cdot\mathbf{step}(C_t,\sigma)(C_t',\sigma')\cdot\\
&\qquad \delta(C_1)(C_1')\cdots\delta(C_{t-1})(C_{t-1}')\cdot\delta(C_{t+1})(C_{t+1}')\cdots\delta(C_n)(C_n')\\
&= \lambda(C_1' \parallel \ldots C_n',\sigma').\sum_{C_1,\ldots,C_n,\sigma} W(C_1\parallel\cdots\parallel C_n,\sigma)\cdot\\
&\qquad (\delta(C_1)\parallel\cdots\parallel\delta(C_{t-1})\parallel\mathbf{step}(C_t,\sigma)\parallel\delta(C_{t+1})\parallel\cdots\parallel\delta(C_n))(C_1'\parallel\cdots\parallel C_n',\sigma')\\
&= \lambda(C_1' \parallel \ldots C_n',\sigma').\sum_{C_1,\ldots,C_n,\sigma} W(C_1\parallel\cdots\parallel C_n,\sigma)\cdot\\
&\qquad \mathbf{step}(C_1\parallel\cdots\parallel C_n,\sigma,t)(C_1'\parallel\cdots\parallel C_n',\sigma')\\
&= \lambda(\mathbb{C}',\sigma').\sum_{\mathbb{C},\sigma} W(\mathbb{C},\sigma)\cdot\mathbf{step}(\mathbb{C},\sigma,t)(\mathbb{C}',\sigma')\\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{step}(\mathbb{C},\sigma,t)\}\\
&= \mathbf{step}(W,t).
\end{aligned}$$

Lemma 110. *For all W, t, W', $W \overset{t}{\rightsquigarrow} W'$ if and only if $W' = \boldsymbol{step}(W,t)$.*

Proof. For all W, t, W', we prove the two directions respectively.

- if $W \overset{t}{\rightsquigarrow} W'$, then $W' = \mathbf{step}(W,t)$.

  By Lemma. 109.
- if $W' = \mathbf{step}(W,t)$, then $W \overset{t}{\rightsquigarrow} W'$.

  Let $W'' \overset{\text{def}}{=} \lambda(\mathbb{C}',\sigma').\sum_{\mathbb{C},\sigma}\{W(\mathbb{C},\sigma)\cdot p \mid (\mathbb{C},\sigma)\xrightarrow[t]{p}(\mathbb{C}',\sigma')\}$, then $W \overset{t}{\rightsquigarrow} W''$.

  By Lem. 109 we know $W'' = \mathbf{step}(W,t)$, so $W \overset{t}{\rightsquigarrow} \mathbf{step}(W,t)$.

Lemma 111. *For all* W, f, t, $\boldsymbol{step}(\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{f(\mathbb{C},\sigma)\}, t) = \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\boldsymbol{step}(f(\mathbb{C},\sigma), t)\}$.

Proof. For all W, f, t, we have

$$\begin{aligned}
&\mathbf{step}(\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{f(\mathbb{C},\sigma)\}, t)\\
&= \mathbb{E}_{(\mathbb{C}',\sigma')\sim\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{f(\mathbb{C},\sigma)\}}\{\mathbf{step}(\mathbb{C}',\sigma',t)\}\\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbb{E}_{(\mathbb{C}',\sigma')\sim f(\mathbb{C},\sigma)}\{\mathbf{step}(\mathbb{C}',\sigma',t)\}\} \qquad \text{(by Lem. 15)}\\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{step}(f(\mathbb{C},\sigma), t)\}.
\end{aligned}$$

Lemma 112. *For all* W, f, $\mathbf{splitAtom}(\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{f(\mathbb{C},\sigma)\}) = \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{splitAtom}(f(\mathbb{C},\sigma))\}$.

Proof. For all W, f, we have

$$\begin{aligned}
&\mathbf{splitAtom}(\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{f(\mathbb{C},\sigma)\})\\
&= \mathbb{E}_{(\mathbb{C}',\sigma')\sim\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{f(\mathbb{C},\sigma)\}}\{\mathbf{splitAtom}(\mathbb{C}')\otimes\delta(\sigma')\}\\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbb{E}_{(\mathbb{C}',\sigma')\sim f(\mathbb{C},\sigma)}\{\mathbf{splitAtom}(\mathbb{C}')\otimes\delta(\sigma')\}\} \qquad \text{(by Lem. 15)}\\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{splitAtom}(f(\mathbb{C},\sigma))\}.
\end{aligned}$$

Lemma 113. *For all* W, f, $\mathbf{lazycoin}(\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{f(\mathbb{C},\sigma)\}) = \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{lazycoin}(f(\mathbb{C},\sigma))\}$.

Proof. For all W, f, we have

$$\begin{aligned}
&\mathbf{lazycoin}(\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{f(\mathbb{C},\sigma)\})\\
&= \mathbb{E}_{(\mathbb{C}',\sigma')\sim\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{f(\mathbb{C},\sigma)\}}\{\delta(\mathbf{lazycoin}(\mathbb{C}'))\otimes\delta(\sigma')\}\\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbb{E}_{(\mathbb{C}',\sigma')\sim f(\mathbb{C},\sigma)}\{\delta(\mathbf{lazycoin}(\mathbb{C}'))\otimes\delta(\sigma')\}\} \qquad \text{(by Lem. 15)}\\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{lazycoin}(f(\mathbb{C},\sigma))\}.
\end{aligned}$$

Lemma 114. *For all* $\rho_1, \ldots, \rho_{t-1}, \eta, \rho_{t+1}, \ldots, \rho_n$, $\mathbf{splitAtom}(\rho_1 \parallel \cdots \parallel \rho_{t-1} \parallel \eta \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n) = \mathbf{splitAtom}(\rho_1) \parallel \cdots \parallel \mathbf{splitAtom}(\rho_{t-1}) \parallel \mathbf{splitAtom}(\eta) \parallel \mathbf{splitAtom}(\rho_{t+1}) \parallel \cdots \parallel \mathbf{splitAtom}(\rho_n)$.

Proof. For all $\rho_1, \ldots, \rho_{t-1}, \eta, \rho_{t+1}, \ldots, \rho_n$,

$$\begin{aligned}
&\mathbf{splitAtom}(\rho_1 \parallel \cdots \parallel \rho_{t-1} \parallel \eta \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n)\\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim\rho_1\parallel\cdots\parallel\rho_{t-1}\parallel\eta\parallel\rho_{t+1}\parallel\cdots\parallel\rho_n}\{\mathbf{splitAtom}(\mathbb{C})\otimes\delta(\sigma)\}\\
&= \lambda(\mathbb{C}',\sigma').\textstyle\sum_{\mathbb{C},\sigma}(\rho_1 \parallel \cdots \parallel \rho_{t-1} \parallel \eta \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n)(\mathbb{C},\sigma)\cdot\mathbf{splitAtom}(\mathbb{C})(\mathbb{C}')\cdot\delta(\sigma)(\sigma')\\
&= \lambda(C_1' \parallel \cdots \parallel C_n',\sigma').\textstyle\sum_{C_1,\ldots,C_n,\sigma}(\rho_1 \parallel \cdots \parallel \rho_{t-1} \parallel \eta \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n)(C_1 \parallel \cdots \parallel C_n,\sigma)\cdot\\
&\qquad \mathbf{splitAtom}(C_1 \parallel \cdots \parallel C_n)(C_1' \parallel \cdots \parallel C_n')\cdot\delta(\sigma)(\sigma')\\
&= \lambda(C_1' \parallel \cdots \parallel C_n',\sigma').\textstyle\sum_{C_1,\ldots,C_n,\sigma}\rho_1(C_1)\cdots\rho_{t-1}(C_{t-1})\cdot\eta(C_t,\sigma)\cdot\rho_{t+1}(C_{t+1})\cdots\rho_n(C_n)\cdot\\
&\qquad \mathbf{splitAtom}(C_1)(C_1')\cdots\mathbf{splitAtom}(C_n)(C_n')\cdot\delta(\sigma)(\sigma')\\
&= \lambda(C_1' \parallel \cdots \parallel C_n',\sigma').\ (\textstyle\sum_{C_1}\rho_1(C_1)\cdot\mathbf{splitAtom}(C_1)(C_1'))\cdots\\
&\qquad (\textstyle\sum_{C_{t-1}}\rho_{t-1}(C_{t-1})\cdot\mathbf{splitAtom}(C_{t-1})(C_{t-1}'))\cdot(\textstyle\sum_{C_t,\sigma}\eta(C_t,\sigma)\cdot\mathbf{splitAtom}(C_t)(C_t')\cdot\delta(\sigma)(\sigma'))\cdot\\
&\qquad (\textstyle\sum_{C_{t+1}}\rho_{t+1}(C_{t+1})\cdot\mathbf{splitAtom}(C_{t+1})(C_{t+1}'))\cdots(\textstyle\sum_{C_n}\rho_n(C_n)\cdot\mathbf{splitAtom}(C_n)(C_n'))\\
&= \lambda(C_1' \parallel \cdots \parallel C_n',\sigma').\ \mathbb{E}_{C_1\sim\rho_1}\{\mathbf{splitAtom}(C_1)\}(C_1')\cdots\mathbb{E}_{C_{t-1}\sim\rho_{t-1}}\{\mathbf{splitAtom}(C_{t-1})\}(C_{t-1}')\cdot\\
&\qquad \mathbb{E}_{(C_t,\sigma)\sim\eta}\{\mathbf{splitAtom}(C_t)\otimes\delta(\sigma)\}(C_t',\sigma')\cdot\mathbb{E}_{C_{t+1}\sim\rho_{t+1}}\{\mathbf{splitAtom}(C_{t+1})\}(C_{t+1}')\cdots\\
&\qquad \mathbb{E}_{C_n\sim\rho_n}\{\mathbf{splitAtom}(C_n)\}(C_n')\\
&= \lambda(C_1' \parallel \cdots \parallel C_n',\sigma').\ \mathbf{splitAtom}(\rho_1)(C_1')\cdots\mathbf{splitAtom}(\rho_{t-1})(C_{t-1}')\cdot\mathbf{splitAtom}(\eta)(C_t',\sigma')\cdot\\
&\qquad \mathbf{splitAtom}(\rho_{t+1})(C_{t+1}')(C_{t+1}')\cdots\mathbf{splitAtom}(\rho_n)(C_n')(C_n')\\
&= \mathbf{splitAtom}(\rho_1) \parallel \cdots \parallel \mathbf{splitAtom}(\rho_{t-1}) \parallel \mathbf{splitAtom}(\eta) \parallel \mathbf{splitAtom}(\rho_{t+1}) \parallel \cdots \parallel \mathbf{splitAtom}(\rho_n).
\end{aligned}$$

Lemma 115. *For all* $\rho_1, \ldots, \rho_{t-1}, \eta, \rho_{t+1}, \ldots, \rho_n$, $\mathbf{lazycoin}(\rho_1 \parallel \ldots \rho_{t-1} \parallel \eta \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n) = \mathbf{lazycoin}(\rho_1) \parallel \ldots \mathbf{lazycoin}(\rho_{t-1}) \parallel \mathbf{lazycoin}(\eta) \parallel \mathbf{lazycoin}(\rho_{t+1}) \parallel \cdots \parallel \mathbf{lazycoin}(C_n)$.

Proof. For all $\rho_1, \ldots, \rho_{t-1}, \eta, \rho_{t+1}, \ldots, \rho_n$,

$$
\begin{aligned}
&\mathbf{lazycoin}(\rho_1 \parallel \cdots \parallel \rho_{t-1} \parallel \eta \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n) \\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim\rho_1\parallel\cdots\parallel\rho_{t-1}\parallel\eta\parallel\rho_{t+1}\parallel\cdots\parallel\rho_n}\{\delta(\mathbf{lazycoin}(\mathbb{C})) \otimes \delta(\sigma)\} \\
&= \lambda(\mathbb{C}', \sigma').\, \textstyle\sum_{\mathbb{C},\sigma}(\rho_1 \parallel \cdots \parallel \rho_{t-1} \parallel \eta \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n)(\mathbb{C}, \sigma) \cdot \delta(\mathbf{lazycoin}(\mathbb{C}))(\mathbb{C}') \cdot \delta(\sigma)(\sigma') \\
&= \lambda(C_1' \parallel \cdots \parallel C_n', \sigma').\, \textstyle\sum_{C_1,\ldots,C_n,\sigma}(\rho_1 \parallel \cdots \parallel \rho_{t-1} \parallel \eta \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n)(C_1 \parallel \cdots \parallel C_n, \sigma) \cdot \\
&\qquad \delta(\mathbf{lazycoin}(C_1 \parallel \cdots \parallel C_n))(C_1' \parallel \cdots \parallel C_n') \cdot \delta(\sigma)(\sigma') \\
&= \lambda(C_1' \parallel \cdots \parallel C_n', \sigma').\, \textstyle\sum_{C_1,\ldots,C_n,\sigma} \rho_1(C_1) \cdots \rho_{t-1}(C_{t-1}) \cdot \eta(C_t, \sigma) \cdot \rho_{t+1}(C_{t+1}) \cdots \rho_n(C_n) \cdot \\
&\qquad \delta(\mathbf{lazycoin}(C_1))(C_1') \cdots \delta(\mathbf{lazycoin}(C_n))(C_n') \cdot \delta(\sigma)(\sigma') \\
&= \lambda(C_1' \parallel \cdots \parallel C_n', \sigma').\, (\textstyle\sum_{C_1} \rho_1(C_1) \cdot \delta(\mathbf{lazycoin}(C_1))(C_1')) \cdots \\
&\qquad (\textstyle\sum_{C_{t-1}} \rho_{t-1}(C_{t-1}) \cdot \delta(\mathbf{lazycoin}(C_{t-1}))(C_{t-1}')) \cdot (\textstyle\sum_{C_t,\sigma} \eta(C_t, \sigma) \cdot \delta(\mathbf{lazycoin}(C_t))(C_t') \cdot \delta(\sigma)(\sigma')) \cdot \\
&\qquad (\textstyle\sum_{C_{t+1}} \rho_{t+1}(C_{t+1}) \cdot \delta(\mathbf{lazycoin}(C_{t+1}))(C_{t+1}')) \cdots (\textstyle\sum_{C_n} \rho_n(C_n) \cdot \delta(\mathbf{lazycoin}(C_n))(C_n')) \\
&= \lambda(C_1' \parallel \cdots \parallel C_n', \sigma').\, \mathbb{E}_{C_1\sim\rho_1}\{\delta(\mathbf{lazycoin}(C_1))\}(C_1') \cdots \mathbb{E}_{C_{t-1}\sim\rho_{t-1}}\{\delta(\mathbf{lazycoin}(C_{t-1}))\}(C_{t-1}') \cdot \\
&\qquad \mathbb{E}_{(C_t,\sigma)\sim\eta}\{\delta(\mathbf{lazycoin}(C_t)) \otimes \delta(\sigma)\}(C_t', \sigma') \cdot \mathbb{E}_{C_{t+1}\sim\rho_{t+1}}\{\delta(\mathbf{lazycoin}(C_{t+1}))\}(C_{t+1}') \cdots \\
&\qquad \mathbb{E}_{C_n\sim\rho_n}\{\delta(\mathbf{lazycoin}(C_n))\}(C_n') \\
&= \lambda(C_1' \parallel \cdots \parallel C_n', \sigma').\, \mathbf{lazycoin}(\rho_1)(C_1') \cdots \mathbf{lazycoin}(\rho_{t-1})(C_{t-1}') \cdot \mathbf{lazycoin}(\eta)(C_t', \sigma') \cdot \\
&\qquad \mathbf{lazycoin}(\rho_{t+1})(C_{t+1}')(C_{t+1}') \cdots \mathbf{lazycoin}(\rho_n)(C_n')(C_n') \\
&= \mathbf{lazycoin}(\rho_1) \parallel \cdots \parallel \mathbf{lazycoin}(\rho_{t-1}) \parallel \mathbf{lazycoin}(\eta) \parallel \mathbf{lazycoin}(\rho_{t+1}) \parallel \cdots \parallel \mathbf{lazycoin}(\rho_n).
\end{aligned}
$$

Lemma 116. *For all* ρ, μ, $\mathbf{splitAtom}(\rho \otimes \mu) = \mathbf{splitAtom}(\rho) \otimes \mu$.

Proof. For all ρ, μ,

$$
\begin{aligned}
&\mathbf{splitAtom}(\rho \otimes \mu) \\
&= \mathbb{E}_{(C,\sigma)\sim\rho\otimes\mu}\{\mathbf{splitAtom}(C) \otimes \delta(\sigma)\} \\
&= \lambda(C', \sigma').\, \textstyle\sum_{C,\sigma} \rho(C) \cdot \mu(\sigma) \cdot \mathbf{splitAtom}(C)(C') \cdot \delta(\sigma)(\sigma') \\
&= \lambda(C', \sigma').\, (\textstyle\sum_{C} \rho(C) \cdot \mathbf{splitAtom}(C)(C')) \cdot \mu(\sigma') \\
&= \lambda(C', \sigma').\, \mathbb{E}_{C\sim\rho}\{\mathbf{splitAtom}(C)\}(C') \cdot \mu(\sigma') \\
&= \lambda(C', \sigma').\, \mathbf{splitAtom}(\rho)(C') \cdot \mu(\sigma') \\
&= \mathbf{splitAtom}(\rho) \otimes \mu.
\end{aligned}
$$

Lemma 117. *For all* ρ, μ, $\mathbf{lazycoin}(\rho \otimes \mu) = \mathbf{lazycoin}(\rho) \otimes \mu$.

Proof. For all ρ, μ,

$$
\begin{aligned}
&\mathbf{lazycoin}(\rho \otimes \mu) \\
&= \mathbb{E}_{(C,\sigma)\sim\rho\otimes\mu}\{\delta(\mathbf{lazycoin}(C)) \otimes \delta(\sigma)\} \\
&= \lambda(C', \sigma').\, \textstyle\sum_{C,\sigma} \rho(C) \cdot \mu(\sigma) \cdot \delta(\mathbf{lazycoin}(C))(C') \cdot \delta(\sigma)(\sigma') \\
&= \lambda(C', \sigma').\, (\textstyle\sum_{C} \rho(C) \cdot \delta(\mathbf{lazycoin}(C))(C')) \cdot \mu(\sigma') \\
&= \lambda(C', \sigma').\, \mathbb{E}_{C\sim\rho}\{\delta(\mathbf{lazycoin}(C))\}(C') \cdot \mu(\sigma') \\
&= \lambda(C', \sigma').\, \mathbf{lazycoin}(\rho)(C') \cdot \mu(\sigma') \\
&= \mathbf{lazycoin}(\rho) \otimes \mu.
\end{aligned}
$$

Lemma 118. *For all* C, $\mathbf{splitAtom}(\delta(C)) = \mathbf{splitAtom}(C)$.

Proof. For all C,

$$\begin{aligned}&\mathbf{splitAtom}(\delta(C))\\ =\ &\mathbb{E}_{C'\sim\delta(C)}\{\mathbf{splitAtom}(C')\}\\ =\ &\mathbf{splitAtom}(C). \qquad \text{(by Lem. 17)}\end{aligned}$$

Lemma 119. *For all* C, $\mathbf{lazycoin}(\delta(C)) = \delta(\mathbf{lazycoin}(C))$.

Proof. For all C,

$$\begin{aligned}&\mathbf{lazycoin}(\delta(C))\\ =\ &\mathbb{E}_{C'\sim\delta(C)}\{\delta(\mathbf{lazycoin}(C'))\}\\ =\ &\delta(\mathbf{lazycoin}(C)). \qquad \text{(by Lem. 17)}\end{aligned}$$

Lemma 120. *For all* C *and* σ, $\boldsymbol{step}(\delta(C)\otimes\delta(\sigma)) = \boldsymbol{step}(C,\sigma)$.

Proof. For all C and σ,

$$\begin{aligned}&\mathbf{step}(\delta(C)\otimes\delta(\sigma))\\ =\ &\mathbb{E}_{(C',\sigma')\sim\delta(C)\otimes\delta(\sigma)}\{\mathbf{step}(C',\sigma')\}\\ =\ &\lambda(C'',\sigma'').\textstyle\sum_{C',\sigma'}\delta(C)(C')\cdot\delta(\sigma)(\sigma')\cdot\mathbf{step}(C',\sigma')(C'',\sigma'')\\ =\ &\lambda(C'',\sigma'').\mathbf{step}(C,\sigma)(C'',\sigma'').\\ =\ &\mathbf{step}(C,\sigma).\end{aligned}$$

Lemma 121. *For all* $\mathbb{C},\sigma,t$, $\boldsymbol{step}(\delta(\mathbb{C})\otimes\delta(\sigma),t) = \boldsymbol{step}(\mathbb{C},\sigma,t)$.

Proof. For all $\mathbb{C},\sigma,t$,

$$\begin{aligned}&\mathbf{step}(\delta(\mathbb{C})\otimes\delta(\sigma),t)\\ =\ &\mathbb{E}_{(\mathbb{C}',\sigma')\sim\delta(\mathbb{C})\otimes\delta(\sigma)}\{\mathbf{step}(\mathbb{C}',\sigma',t)\}\\ =\ &\lambda(\mathbb{C}'',\sigma'').\textstyle\sum_{\mathbb{C}',\sigma'}\delta(\mathbb{C})(\mathbb{C}')\cdot\delta(\sigma)(\sigma')\cdot\mathbf{step}(\mathbb{C}',\sigma',t)(\mathbb{C}'',\sigma'').\\ =\ &\lambda(\mathbb{C}'',\sigma'').\mathbf{step}(\mathbb{C},\sigma)(\mathbb{C}'',\sigma'').\\ =\ &\mathbf{step}(\mathbb{C},\sigma,t).\end{aligned}$$

Lemma 122. *For all* η_1,η_2,p, $\mathbf{splitAtom}(\eta_1\oplus_p\eta_2) = \mathbf{splitAtom}(\eta_1)\oplus_p\mathbf{splitAtom}(\eta_2)$.

Proof. For all η_1,η_2,p,

$$\begin{aligned}&\mathbf{splitAtom}(\eta_1\oplus_p\eta_2)\\ =\ &\mathbb{E}_{(\mathbb{C},\sigma)\sim\eta_1\oplus_p\eta_2}\{\mathbf{splitAtom}(\mathbb{C})\otimes\delta(\sigma)\}\\ =\ &\mathbb{E}_{(\mathbb{C},\sigma)\sim\eta_1}\{\mathbf{splitAtom}(\mathbb{C})\otimes\delta(\sigma)\}\oplus_p\mathbb{E}_{(\mathbb{C},\sigma)\sim\eta_2}\{\mathbf{splitAtom}(\mathbb{C})\otimes\delta(\sigma)\} \qquad \text{(by Lem. 16)}\\ =\ &\mathbf{splitAtom}(\eta_1)\oplus_p\mathbf{splitAtom}(\eta_2).\end{aligned}$$

Lemma 123. *For all* η_1,η_2,p, $\mathbf{lazycoin}(\eta_1\oplus_p\eta_2) = \mathbf{lazycoin}(\eta_1)\oplus_p\mathbf{lazycoin}(\eta_2)$.

Proof. For all η_1,η_2,p,

$$\begin{aligned}&\mathbf{lazycoin}(\eta_1\oplus_p\eta_2)\\ =\ &\mathbb{E}_{(\mathbb{C},\sigma)\sim\eta_1\oplus_p\eta_2}\{\delta(\mathbf{lazycoin}(\mathbb{C}))\otimes\delta(\sigma)\}\\ =\ &\mathbb{E}_{(\mathbb{C},\sigma)\sim\eta_1}\{\delta(\mathbf{lazycoin}(\mathbb{C}))\otimes\delta(\sigma)\}\oplus_p\mathbb{E}_{(\mathbb{C},\sigma)\sim\eta_2}\{\delta(\mathbf{lazycoin}(\mathbb{C}))\otimes\delta(\sigma)\} \qquad \text{(by Lem. 16)}\\ =\ &\mathbf{lazycoin}(\eta_1)\oplus_p\mathbf{lazycoin}(\eta_2).\end{aligned}$$

Lemma 124. *For all* η_1, η_2, p, $\boldsymbol{step}(\eta_1 \oplus_p \eta_2) = \boldsymbol{step}(\eta_1) \oplus_p \boldsymbol{step}(\eta_2)$.

Proof. For all η_1, η_2, p,

$$
\begin{aligned}
&\mathbf{splitAtom}(\eta_1 \oplus_p \eta_2) \\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim\eta_1\oplus_p\eta_2}\{\mathbf{splitAtom}(\mathbb{C}) \otimes \delta(\sigma)\} \\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim\eta_1}\{\mathbf{splitAtom}(\mathbb{C}) \otimes \delta(\sigma)\} \oplus_p \mathbb{E}_{(\mathbb{C},\sigma)\sim\eta_2}\{\mathbf{splitAtom}(\mathbb{C}) \otimes \delta(\sigma)\} \qquad \text{(by Lem. 16)} \\
&= \mathbf{splitAtom}(\eta_1) \oplus_p \mathbf{splitAtom}(\eta_2).
\end{aligned}
$$

Lemma 125. *For all* C_1, C_2, σ, *if* $C_1 \neq \boldsymbol{skip}$, *then* $\boldsymbol{step}(C_1; C_2, \sigma) = \boldsymbol{step}(C_1, \sigma); C_2$.

Proof. For all C_1, C_2, σ such that $C_1 \neq \mathbf{skip}$, we have

$$
\begin{aligned}
&\mathbf{step}(C_1; C_2, \sigma) \\
&= \lambda(C', \sigma').\begin{cases} p, & \text{if } (C_1; C_2, \sigma) \xrightarrow{p} (C', \sigma') \\ 0, & \text{otherwise} \end{cases} \\
&= \lambda(C', \sigma').\begin{cases} p, & \text{if } (C_1, \sigma) \xrightarrow{p} (C_1', \sigma') \wedge C' = C_1'; C_2 \\ 0, & \text{otherwise} \end{cases} \\
&= \lambda(C', \sigma').\begin{cases} \mathbf{step}(C_1, \sigma)(C_1', \sigma'), & \text{if } C' = C_1'; C_2 \\ 0, & \text{otherwise} \end{cases} \\
&= \mathbf{step}(C_1, \sigma); C_2.
\end{aligned}
$$

Lemma 126. *For all* η, C_2, $\mathbf{splitAtom}(\eta; C_2) = \mathbf{splitAtom}(\eta); C_2$.

Proof. For all η, C_2,

$$
\begin{aligned}
&\mathbf{splitAtom}(\eta; C_2) \\
&= \mathbb{E}_{(C,\sigma)\sim\eta;C_2}\{\mathbf{splitAtom}(C) \otimes \delta(\sigma)\} \\
&= \lambda(C', \sigma').\textstyle\sum_{C,\sigma}(\eta; C_2)(C, \sigma) \cdot \mathbf{splitAtom}(C)(C') \cdot \delta(\sigma)(\sigma') \\
&= \lambda(C', \sigma').\textstyle\sum_{C_1,\sigma}\eta(C_1, \sigma) \cdot \mathbf{splitAtom}(C_1; C_2)(C') \cdot \delta(\sigma)(\sigma') \\
&= \lambda(C', \sigma').\textstyle\sum_{C_1,\sigma}\eta(C_1, \sigma) \cdot (\mathbf{splitAtom}(C_1); C_2)(C') \cdot \delta(\sigma)(\sigma') \\
&= \lambda(C', \sigma').\begin{cases} \sum_{C_1,\sigma}\eta(C_1, \sigma) \cdot \mathbf{splitAtom}(C_1)(C_1') \cdot \delta(\sigma)(\sigma'), & \text{if } C' = C_1'; C_2 \\ 0, & \text{otherwise} \end{cases} \\
&= \lambda(C', \sigma').\begin{cases} \mathbb{E}_{(C_1,\sigma)\sim\eta}\{\mathbf{splitAtom}(C_1) \otimes \delta(\sigma)\}(C_1', \sigma'), & \text{if } C' = C_1'; C_2 \\ 0, & \text{otherwise} \end{cases} \\
&= \lambda(C', \sigma').\begin{cases} \mathbf{splitAtom}(\eta)(C_1', \sigma'), & \text{if } C' = C_1'; C_2 \\ 0, & \text{otherwise} \end{cases} \\
&= \mathbf{splitAtom}(\eta); C_2.
\end{aligned}
$$

Lemma 127. *For all* η, C_2, $\mathbf{lazycoin}(\eta; C_2) = \mathbf{lazycoin}(\eta); \mathbf{lazycoin}(C_2)$.

Proof.

$$
\begin{aligned}
&\mathbf{lazycoin}(\eta; C_2)\\
&= \mathbb{E}_{(C,\sigma)\sim\eta;C_2}\{\delta(\mathbf{lazycoin}(C)) \otimes \delta(\sigma)\}\\
&= \lambda(C',\sigma').\textstyle\sum_{C,\sigma}(\eta; C_2)(C,\sigma)\cdot\delta(\mathbf{lazycoin}(C))(C')\cdot\delta(\sigma)(\sigma')\\
&= \lambda(C',\sigma').\textstyle\sum_{C_1,\sigma}\eta(C_1,\sigma)\cdot\delta(\mathbf{lazycoin}(C_1;C_2))(C')\cdot\delta(\sigma)(\sigma')\\
&= \lambda(C',\sigma').\textstyle\sum_{C_1,\sigma}\eta(C_1,\sigma)\cdot\delta(\mathbf{lazycoin}(C_1);\mathbf{lazycoin}(C_2))(C')\cdot\delta(\sigma)(\sigma')\\
&= \lambda(C',\sigma').\begin{cases}\sum_{C_1,\sigma}\eta(C_1,\sigma)\cdot\delta(\mathbf{lazycoin}(C_1))(C_1')\cdot\delta(\sigma)(\sigma'), & \text{if } C' = C_1';\mathbf{lazycoin}(C_2)\\ 0, & \text{otherwise}\end{cases}\\
&= \lambda(C',\sigma').\begin{cases}\mathbb{E}_{(C_1,\sigma)\sim\eta}\{\delta(\mathbf{lazycoin}(C_1))\otimes\delta(\sigma)\}(C_1',\sigma'), & \text{if } C' = C_1';C_2\\ 0, & \text{otherwise}\end{cases}\\
&= \lambda(C',\sigma').\begin{cases}\mathbf{lazycoin}(\eta)(C_1',\sigma'), & \text{if } C' = C_1';C_2\\ 0, & \text{otherwise}\end{cases}\\
&= \mathbf{lazycoin}(\eta);C_2.
\end{aligned}
$$

Lemma 128. *For all* C,σ, $\mathbf{splitAtom}(\mathbf{\mathit{step}}(\mathbf{lazycoin}(C),\sigma)) = \mathbf{lazycoin}(\mathbf{\mathit{step}}(C,\sigma))$.

Proof. by induction on C.

- case 1: $C = \langle C_1\rangle \oplus_p \langle C_2\rangle$.
  For all σ, we have

$$
\begin{aligned}
&\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C),\sigma))\\
&= \mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(\langle C_1\rangle \oplus_p \langle C_2\rangle),\sigma))\\
&= \mathbf{splitAtom}(\mathbf{step}(\mathbf{skip};\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle,\sigma))\\
&= \mathbf{splitAtom}(\delta(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle)\otimes\delta(\sigma))\\
&= \mathbf{splitAtom}(\delta(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle))\otimes\delta(\sigma) \qquad \text{(by Lem. 116)}\\
&= \mathbf{splitAtom}(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle)\otimes\delta(\sigma) \qquad \text{(by Lem. 118)}\\
&= \delta(\langle C_1\rangle)\oplus_p\delta(\langle C_2\rangle)\otimes\delta(\sigma)\\
&= (\delta(\langle C_1\rangle)\otimes\delta(\sigma))\oplus_p(\delta(\langle C_2\rangle)\otimes\delta(\sigma)). \qquad \text{(by Lem. 14)}
\end{aligned}
$$

  and

$$
\begin{aligned}
&\mathbf{lazycoin}(\mathbf{step}(C,\sigma))\\
&= \mathbf{lazycoin}(\mathbf{step}(\langle C_1\rangle \oplus_p \langle C_2\rangle,\sigma))\\
&= \mathbf{lazycoin}((\delta(\langle C_1\rangle)\otimes\delta(\sigma))\oplus_p(\delta(\langle C_2\rangle)\otimes\delta(\sigma)))\\
&= \mathbf{lazycoin}(\delta(\langle C_1\rangle)\otimes\delta(\sigma))\oplus_p\mathbf{lazycoin}(\delta(\langle C_2\rangle)\otimes\delta(\sigma)) \qquad \text{(by Lem. 123)}\\
&= (\mathbf{lazycoin}(\delta(\langle C_1\rangle))\otimes\delta(\sigma))\oplus_p(\mathbf{lazycoin}(\delta(\langle C_2\rangle))\otimes\delta(\sigma)) \qquad \text{(by Lem. 117)}\\
&= (\delta(\mathbf{lazycoin}(\langle C_1\rangle))\otimes\delta(\sigma))\oplus_p(\delta(\mathbf{lazycoin}(\langle C_2\rangle))\otimes\delta(\sigma)) \qquad \text{(by Lem. 119)}\\
&= (\delta(\langle C_1\rangle)\otimes\delta(\sigma))\oplus_p(\delta(\langle C_2\rangle)\otimes\delta(\sigma)).
\end{aligned}
$$

  Therefore $\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C),\sigma)) = \mathbf{lazycoin}(\mathbf{step}(C,\sigma))$.
- case 2: $C = C_1;C_2$.
  IH: for all σ, $\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C_1),\sigma)) = \mathbf{lazycoin}(\mathbf{step}(C_1,\sigma))$.
  For all σ, we have $\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C),\sigma)) = \mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C_1;C_2),\sigma))$
  $= \mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C_1);\mathbf{lazycoin}(C_2),\sigma))$. It is obvious that $C_1 =$
  $\mathbf{skip}$ or $C_1 \neq \mathbf{skip}$, we prove the two cases respectively.

- $C_1 = \mathbf{skip}$.
  We have

$$\begin{aligned}
&\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C), \sigma)) \\
=\ &\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(\mathbf{skip}); \mathbf{lazycoin}(C_2), \sigma)) \\
=\ &\mathbf{splitAtom}(\mathbf{step}(\mathbf{skip}; \mathbf{lazycoin}(C_2), \sigma)) \\
=\ &\mathbf{splitAtom}(\delta(\mathbf{lazycoin}(C_2)) \otimes \delta(\sigma)) \\
=\ &\mathbf{splitAtom}(\delta(\mathbf{lazycoin}(C_2))) \otimes \delta(\sigma) \qquad \text{(by Lem. 116)} \\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(C_2)) \otimes \delta(\sigma) \qquad \text{(by Lem. 118)} \\
=\ &\delta(\mathbf{lazycoin}(C_2)) \otimes \delta(\sigma) \qquad \text{(by Lem. 97)}
\end{aligned}$$

  and

$$\begin{aligned}
&\mathbf{lazycoin}(\mathbf{step}(C, \sigma)) \\
=\ &\mathbf{lazycoin}(\mathbf{step}(\mathbf{skip}; C_2, \sigma)) \\
=\ &\mathbf{lazycoin}(\delta(C_2) \otimes \delta(\sigma)) \\
=\ &\mathbf{lazycoin}(\delta(C_2)) \otimes \delta(\sigma) \qquad \text{(by Lem. 117)} \\
=\ &\delta(\mathbf{lazycoin}(C_2)) \otimes \delta(\sigma). \qquad \text{(by Lem. 119)}
\end{aligned}$$

  Therefore $\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C), \sigma)) = \mathbf{lazycoin}(\mathbf{step}(C, \sigma))$.
- $C_1 \neq \mathbf{skip}$.
  From $C_1 \neq \mathbf{skip}$ we know $\mathbf{lazycoin}(C_1) \neq \mathbf{skip}$.
  By Lem. 125 we know $\mathbf{step}(C_1; C_2, \sigma) = \mathbf{step}(C_1, \sigma); C_2$ and $\mathbf{step}(\mathbf{lazycoin}(C_1); \mathbf{lazycoin}(C_2), \sigma) = \mathbf{step}(\mathbf{lazycoin}(C_1), \sigma); \mathbf{lazycoin}(C_2)$, thus

$$\begin{aligned}
&\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C), \sigma)) \\
=\ &\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C_1); \mathbf{lazycoin}(C_2), \sigma)) \\
=\ &\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C_1), \sigma); \mathbf{lazycoin}(C_2)) \\
=\ &\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C_1), \sigma)); \mathbf{lazycoin}(C_2) \qquad \text{(by Lem. 126)} \\
=\ &\mathbf{lazycoin}(\mathbf{step}(C_1, \sigma)); \mathbf{lazycoin}(C_2) \qquad \text{(by IH)} \\
=\ &\mathbf{lazycoin}(\mathbf{step}(C_1, \sigma); C_2) \qquad \text{(by Lem. 127)} \\
=\ &\mathbf{lazycoin}(\mathbf{step}(C_1; C_2, \sigma)) \\
=\ &\mathbf{lazycoin}(\mathbf{step}(C, \sigma)).
\end{aligned}$$

– case 3: $C = \mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2$.
  For all σ, it is obvious that $\sigma \models b$ or $\sigma \not\models b$. We only prove the case $\sigma \models b$, the other case is similar. From $\sigma \models b$ we know $\mathbf{step}(\mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2) = \delta(C_1) \otimes \delta(\sigma)$ and $\mathbf{step}(\mathbf{if}\ (b)\ \mathbf{then}\ \mathbf{lazycoin}(C_1)\ \mathbf{else}\ \mathbf{lazycoin}(C_2)) = \delta(\mathbf{lazycoin}(C_1)) \otimes \delta(\sigma)$, thus

$$\begin{aligned}
&\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C), \sigma)) \\
=\ &\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(\mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2), \sigma)) \\
=\ &\mathbf{splitAtom}(\mathbf{step}(\mathbf{if}\ (b)\ \mathbf{then}\ \mathbf{lazycoin}(C_1)\ \mathbf{else}\ \mathbf{lazycoin}(C_2), \sigma)) \\
=\ &\mathbf{splitAtom}(\delta(\mathbf{lazycoin}(C_1)) \otimes \delta(\sigma)) \\
=\ &\mathbf{splitAtom}(\delta(\mathbf{lazycoin}(C_1))) \otimes \delta(\sigma) \qquad \text{(by Lem. 116)} \\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(C_1)) \otimes \delta(\sigma) \qquad \text{(by Lem. 118)} \\
=\ &\delta(\mathbf{lazycoin}(C_1)) \otimes \delta(\sigma). \qquad \text{(by Lem. 97)} \\
=\ &\mathbf{lazycoin}(\mathbf{step}(C, \sigma))
\end{aligned}$$

and

$$\begin{aligned}
&\mathbf{lazycoin}(\mathbf{step}(C,\sigma))\\
=\ &\mathbf{lazycoin}(\mathbf{step}(\mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2,\sigma))\\
=\ &\mathbf{lazycoin}(\delta(C_1)\otimes\delta(\sigma))\\
=\ &\mathbf{lazycoin}(\delta(C_1))\otimes\delta(\sigma) \qquad \text{(by Lem. 117)}\\
=\ &\delta(\mathbf{lazycoin}(C_1))\otimes\delta(\sigma). \qquad \text{(by Lem. 119)}
\end{aligned}$$

Therefore $\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C),\sigma)) = \mathbf{lazycoin}(\mathbf{step}(C,\sigma))$.

– case 4: $C = \mathbf{while}\ (b)\ \mathbf{do}\ C_1$.
For all σ, it is obvious that $\sigma \models b$ or $\sigma \not\models b$. We prove the two cases respectively.

- $\sigma \models b$.
We have $\mathbf{step}(\mathbf{while}\ (b)\ \mathbf{do}\ C_1,\sigma) = \delta(C_1;\mathbf{while}\ (b)\ \mathbf{do}\ C_1)\otimes\delta(\sigma)$ and $\mathbf{step}(\mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{lazycoin}(C_1),\sigma) = \delta(\mathbf{lazycoin}(C_1);\mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{lazycoin}(C_1))\otimes\delta(\sigma)$, thus

$$\begin{aligned}
&\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C),\sigma))\\
=\ &\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(\mathbf{while}\ (b)\ \mathbf{do}\ C_1),\sigma))\\
=\ &\mathbf{splitAtom}(\mathbf{step}(\mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{lazycoin}(C_1),\sigma))\\
=\ &\mathbf{splitAtom}(\delta(\mathbf{lazycoin}(C_1);\mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{lazycoin}(C_1))\otimes\delta(\sigma))\\
=\ &\mathbf{splitAtom}(\delta(\mathbf{lazycoin}(C_1);\mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{lazycoin}(C_1)))\otimes\delta(\sigma) \qquad \text{(by Lem. 116)}\\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(C_1);\mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{lazycoin}(C_1))\otimes\delta(\sigma) \qquad \text{(by Lem. 118)}\\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(C_1));\mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{lazycoin}(C_1)\otimes\delta(\sigma) \qquad \text{(by Lem. 126)}\\
=\ &\delta(\mathbf{lazycoin}(C_1));\mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{lazycoin}(C_1)\otimes\delta(\sigma) \qquad \text{(by Lem. 97)}\\
=\ &\delta(\mathbf{lazycoin}(C_1);\mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{lazycoin}(C_1))\otimes\delta(\sigma) \qquad \text{(by Lem. 96)}\\
=\ &\delta(\mathbf{lazycoin}(C_1;\mathbf{while}\ (b)\ \mathbf{do}\ C_1))\otimes\delta(\sigma)
\end{aligned}$$

and

$$\begin{aligned}
&\mathbf{lazycoin}(\mathbf{step}(C,\sigma))\\
=\ &\mathbf{lazycoin}(\mathbf{step}(\mathbf{while}\ (b)\ \mathbf{do}\ C_1,\sigma))\\
=\ &\mathbf{lazycoin}(\delta(C_1;\mathbf{while}\ (b)\ \mathbf{do}\ C_1)\otimes\delta(\sigma))\\
=\ &\mathbf{lazycoin}(\delta(C_1;\mathbf{while}\ (b)\ \mathbf{do}\ C_1))\otimes\delta(\sigma) \qquad \text{(by Lem. 117)}\\
=\ &\delta(\mathbf{lazycoin}(C_1;\mathbf{while}\ (b)\ \mathbf{do}\ C_1))\otimes\delta(\sigma) \qquad \text{(by Lem. 119)}
\end{aligned}$$

Therefore $\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C),\sigma)) = \mathbf{lazycoin}(\mathbf{step}(C,\sigma))$.

- $\sigma \not\models b$.
We have $\mathbf{step}(\mathbf{while}\ (b)\ \mathbf{do}\ C_1,\sigma) = \delta(\mathbf{skip})\otimes\delta(\sigma)$ and $\mathbf{step}(\mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{lazycoin}(C_1),\sigma) = \delta(\mathbf{skip})\otimes\delta(\sigma)$, thus

$$\begin{aligned}
&\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C),\sigma))\\
=\ &\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(\mathbf{while}\ (b)\ \mathbf{do}\ C_1),\sigma))\\
=\ &\mathbf{splitAtom}(\mathbf{step}(\mathbf{while}\ (b)\ \mathbf{do}\ \mathbf{lazycoin}(C_1),\sigma))\\
=\ &\mathbf{splitAtom}(\delta(\mathbf{skip})\otimes\delta(\sigma))\\
=\ &\mathbf{splitAtom}(\delta(\mathbf{skip}))\otimes\delta(\sigma) \qquad \text{(by Lem. 116)}\\
=\ &\mathbf{splitAtom}(\mathbf{skip})\otimes\delta(\sigma) \qquad \text{(by Lem. 118)}\\
=\ &\delta(\mathbf{skip})\otimes\delta(\sigma)
\end{aligned}$$

and

$$\begin{aligned}
&\mathbf{lazycoin}(\mathbf{step}(C,\sigma))\\
=\ &\mathbf{lazycoin}(\mathbf{step}(\mathbf{while}\ (b)\ \mathbf{do}\ C_1,\sigma))\\
=\ &\mathbf{lazycoin}(\delta(\mathbf{skip})\otimes\delta(\sigma))\\
=\ &\mathbf{lazycoin}(\delta(\mathbf{skip}))\otimes\delta(\sigma) \qquad \text{(by Lem. 117)}\\
=\ &\delta(\mathbf{skip})\otimes\delta(\sigma). \qquad \text{(by Lem. 119)}
\end{aligned}$$

Therefore $\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C),\sigma)) = \mathbf{lazycoin}(\mathbf{step}(C,\sigma))$.

– other cases: $C = \mathbf{skip} \mid x := e \mid \langle C_1\rangle \mid \langle C_1\rangle$ *sp*.
We can see $\mathbf{lazycoin}(C) = C$ and there exists μ such that $\mathbf{step}(C,\sigma) = \delta(\mathbf{skip})\otimes\mu$, thus

$$\begin{aligned}
&\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C),\sigma))\\
=\ &\mathbf{splitAtom}(\delta(\mathbf{skip})\otimes\mu)\\
=\ &\mathbf{splitAtom}(\delta(\mathbf{skip}))\otimes\mu \qquad \text{(by Lem. 116)}\\
=\ &\mathbf{splitAtom}(\mathbf{skip})\otimes\mu. \qquad \text{(by Lem. 118)}\\
=\ &\delta(\mathbf{skip})\otimes\mu
\end{aligned}$$

and

$$\begin{aligned}
&\mathbf{lazycoin}(\mathbf{step}(C,\sigma))\\
=\ &\mathbf{lazycoin}(\delta(\mathbf{skip})\otimes\mu\\
=\ &\mathbf{lazycoin}(\delta(\mathbf{skip}))\otimes\mu \qquad \text{(by Lem. 117)}\\
=\ &\delta(\mathbf{skip})\otimes\mu. \qquad \text{(by Lem. 119)}
\end{aligned}$$

Therefore $\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C),\sigma)) = \mathbf{lazycoin}(\mathbf{step}(C,\sigma))$.

Lemma 129. *For all* $\mathbb{C},\sigma,t$, $\mathbf{splitAtom}(\boldsymbol{step}(\mathbf{lazycoin}(\mathbb{C}),\sigma,t)) = \mathbf{lazycoin}(\boldsymbol{step}(\mathbb{C},\sigma,t))$.

Proof. For all $\mathbb{C},\sigma,t$, by definition of *Prog* there exists $C_1,\ldots,C_n$ such that $\mathbb{C} = C_1 \parallel \cdots \parallel C_n$.

$$\begin{aligned}
&\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(\mathbb{C}),\sigma,t))\\
=\ &\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C_1 \parallel \cdots \parallel C_n),\sigma,t))\\
=\ &\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C_1) \parallel \cdots \parallel \mathbf{lazycoin}(C_n),\sigma,t))\\
=\ &\mathbf{splitAtom}(\delta(\mathbf{lazycoin}(C_1)) \parallel \cdots \parallel \delta(\mathbf{lazycoin}(C_{t-1})) \parallel \mathbf{step}(\mathbf{lazycoin}(C_t),\sigma) \parallel\\
&\qquad \delta(\mathbf{lazycoin}(C_{t+1})) \parallel \cdots \parallel \delta(\mathbf{lazycoin}(C_n)))\\
=\ &\mathbf{splitAtom}(\delta(\mathbf{lazycoin}(C_1))) \parallel \cdots \parallel \mathbf{splitAtom}(\delta(\mathbf{lazycoin}(C_{t-1}))) \parallel \mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C_t),\sigma)) \parallel\\
&\mathbf{splitAtom}(\delta(\mathbf{lazycoin}(C_{t+1}))) \parallel \cdots \parallel \mathbf{splitAtom}(\delta(\mathbf{lazycoin}(C_n))) \qquad \text{(by Lem. 114)}\\
=\ &\mathbf{splitAtom}(\mathbf{lazycoin}(C_1)) \parallel \cdots \parallel \mathbf{splitAtom}(\mathbf{lazycoin}(C_{t-1})) \parallel \mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(C_t),\sigma)) \parallel\\
&\mathbf{splitAtom}(\mathbf{lazycoin}(C_{t+1})) \parallel \cdots \parallel \mathbf{splitAtom}(\mathbf{lazycoin}(C_n)) \qquad \text{(by Lem. 118)}\\
=\ &\delta(\mathbf{lazycoin}(C_1)) \parallel \cdots \parallel \delta(\mathbf{lazycoin}(C_{t-1})) \parallel \mathbf{lazycoin}(\mathbf{step}(C_t,\sigma)) \parallel\\
&\delta(\mathbf{lazycoin}(C_{t+1})) \parallel \cdots \parallel \delta(\mathbf{lazycoin}(C_n)) \qquad \text{(by Lem. 97 and Lem. 128)}\\
=\ &\mathbf{lazycoin}(\delta(C_1)) \parallel \cdots \parallel \mathbf{lazycoin}(\delta(C_{t-1})) \parallel \mathbf{lazycoin}(\mathbf{step}(C_t,\sigma)) \parallel\\
&\mathbf{lazycoin}(\delta(C_{t+1})) \parallel \cdots \parallel \mathbf{lazycoin}(\delta(C_n)) \qquad \text{(by Lem. 119)}\\
=\ &\mathbf{lazycoin}(\delta(C_1) \parallel \cdots \parallel \delta(C_{t-1}) \parallel \mathbf{step}(C_t,\sigma) \parallel \delta(C_{t+1}) \parallel \cdots \parallel \delta(C_n)) \qquad \text{(by Lem. 115)}\\
=\ &\mathbf{lazycoin}(\mathbf{step}(C_1 \parallel \cdots \parallel C_n,\sigma,t))\\
=\ &\mathbf{lazycoin}(\mathbf{step}(\mathbb{C},\sigma,t)).
\end{aligned}$$

Lemma 130. *For all* W *and* t, $\mathbf{splitAtom}(\boldsymbol{step}(\mathbf{lazycoin}(W),t)) = \mathbf{lazycoin}(\boldsymbol{step}(W,t))$.

Proof. For all W and t, we have

$$
\begin{aligned}
&\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(W), t)) \\
&= \mathbf{splitAtom}(\mathbf{step}(\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\delta(\mathbf{lazycoin}(\mathbb{C})) \otimes \delta(\sigma)\}, t)) \\
&= \mathbf{splitAtom}(\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{step}(\delta(\mathbf{lazycoin}(\mathbb{C})) \otimes \delta(\sigma), t)\}) \qquad \text{(by Lem. 111)} \\
&= \mathbf{splitAtom}(\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{step}(\mathbf{lazycoin}(\mathbb{C}), \sigma, t)\}) \qquad \text{(by Lem. 121)} \\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(\mathbb{C}), \sigma, t))\} \qquad \text{(by Lem. 112)} \\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{lazycoin}(\mathbf{step}(\mathbb{C}, \sigma, t))\} \qquad \text{(by Lem. 129)} \\
&= \mathbf{lazycoin}(\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{step}(\mathbb{C}, \sigma, t)\}) \qquad \text{(by Lem. 113)} \\
&= \mathbf{lazycoin}(\mathbf{step}(W, t)).
\end{aligned}
$$

Lemma 131. *For all* $\rho_1, \ldots, \rho_n, \mu, t$, $(\rho_1 \parallel \cdots \parallel \rho_n) \otimes \mu = \rho_1 \parallel \ldots \rho_{t-1} \parallel \rho_t \otimes \mu \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n$.

Proof. For all $\rho_1, \ldots, \rho_n, \mu, t$,

$$
\begin{aligned}
&(\rho_1 \parallel \cdots \parallel \rho_n) \otimes \mu \\
&= \lambda(C_1 \parallel \cdots \parallel C_n, \sigma).\ \rho_1(C_1) \cdots \rho_n(C_n) \cdot \mu(\sigma) \\
&= \lambda(C_1 \parallel \cdots \parallel C_n, \sigma).\ \rho_1(C_1) \cdots \rho_{t-1}(C_{t-1}) \cdot (\rho_t(C_t) \cdot \mu(\sigma)) \cdot \rho_{t+1}(C_{t+1}) \cdots \rho_n(C_n) \\
&= \lambda(C_1 \parallel \cdots \parallel C_n, \sigma).\ \rho_1(C_1) \cdots \rho_{t-1}(C_{t-1}) \cdot (\rho_t \otimes \mu)(C_t, \sigma) \cdot \rho_{t+1}(C_{t+1}) \cdots \rho_n(C_n) \\
&= \rho_1 \parallel \ldots \rho_{t-1} \parallel \rho_t \otimes \mu \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n.
\end{aligned}
$$

Lemma 132. *For all* $\rho_1, \ldots, \rho_{t-1}, \eta, \rho_{t+1}, \ldots, \rho_n$, $\boldsymbol{step}(\rho_1 \parallel \cdots \parallel \rho_{t-1} \parallel \eta \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n, t) = \rho_1 \parallel \cdots \parallel \rho_{t-1} \parallel \boldsymbol{step}(\eta) \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n$.

Proof. For all $\rho_1, \ldots, \rho_{t-1}, \eta, \rho_{t+1}, \ldots, \rho_n$,

$$
\begin{aligned}
&\mathbf{step}(\rho_1 \parallel \cdots \parallel \rho_{t-1} \parallel \eta \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n, t) \\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim \rho_1 \parallel \cdots \parallel \rho_{t-1} \parallel \eta \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n}\{\mathbf{step}(\mathbb{C}, \sigma, t)\} \\
&= \lambda(\mathbb{C}', \sigma').\ \textstyle\sum_{\mathbb{C},\sigma}(\rho_1 \parallel \cdots \parallel \rho_{t-1} \parallel \eta \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n)(\mathbb{C}, \sigma) \cdot \mathbf{step}(\mathbb{C}, \sigma, t)(\mathbb{C}', \sigma') \\
&= \lambda(C_1' \parallel \cdots \parallel C_n', \sigma').\ \textstyle\sum_{C_1, \ldots C_n, \sigma} \rho_1(C_1) \cdots \rho_{t-1}(C_{t-1}) \cdot \eta(C_t, \sigma) \cdot \rho_{t+1}(C_{t+1}) \cdots \rho_n(C_n) \cdot \\
&\qquad \mathbf{step}(C_1 \parallel \cdots \parallel C_n, \sigma, t)(C_1' \parallel \cdots \parallel C_n', \sigma') \\
&= \lambda(C_1' \parallel \cdots \parallel C_n', \sigma').\ \textstyle\sum_{C_1, \ldots C_n, \sigma} \rho_1(C_1) \cdots \rho_{t-1}(C_{t-1}) \cdot \eta(C_t, \sigma) \cdot \rho_{t+1}(C_{t+1}) \cdots \rho_n(C_n) \cdot \\
&\qquad \delta(C_1)(C_1') \cdots \delta(C_{t-1})(C_{t-1}') \cdot \mathbf{step}(C_t, \sigma)(C_t', \sigma') \cdot \delta(C_{t+1})(C_{t+1}') \cdots \delta(C_n)(C_n') \\
&= \lambda(C_1' \parallel \cdots \parallel C_n', \sigma').\ \textstyle\sum_{C_t, \sigma} \rho_1(C_1') \cdots \rho_{t-1}(C_{t-1}') \cdot \eta(C_t, \sigma) \cdot \mathbf{step}(C_t, \sigma)(C_t', \sigma') \cdot \\
&\qquad \rho_{t+1}(C_{t+1}') \cdots \rho_n(C_n') \\
&= \lambda(C_1' \parallel \cdots \parallel C_n', \sigma').\rho_1(C_1') \cdots \rho_{t-1}(C_{t-1}') \cdot (\textstyle\sum_{C_t, \sigma} \eta(C_t, \sigma) \cdot \mathbf{step}(C_t, \sigma)(C_t', \sigma')) \cdot \\
&\qquad \rho_{t+1}(C_{t+1}') \cdots \rho_n(C_n') \\
&= \lambda(C_1' \parallel \cdots \parallel C_n', \sigma').\rho_1(C_1') \cdots \rho_{t-1}(C_{t-1}') \cdot \mathbb{E}_{(C_t,\sigma)\sim\eta}\{\mathbf{step}(C_t, \sigma)\}(C_t', \sigma') \cdot \\
&\qquad \rho_{t+1}(C_{t+1}') \cdots \rho_n(C_n') \\
&= \lambda(C_1' \parallel \cdots \parallel C_n', \sigma').\rho_1(C_1') \cdots \rho_{t-1}(C_{t-1}') \cdot \mathbf{step}(\eta)(C_t', \sigma') \cdot \rho_{t+1}(C_{t+1}') \cdots \rho_n(C_n') \\
&= \rho_1 \parallel \cdots \parallel \rho_{t-1} \parallel \mathbf{step}(\eta) \parallel \rho_{t+1} \parallel \cdots \parallel \rho_n.
\end{aligned}
$$

Lemma 133. *For all* C, $\mathbf{splitAtom}(\mathbf{splitAtom}(C)) = \mathbf{splitAtom}(C)$.

Proof. by induction on C.

– case 1: $C = \langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle$.

$$\begin{aligned}
&\mathbf{splitAtom}(\mathbf{splitAtom}(C))\\
=\ &\mathbf{splitAtom}(\mathbf{splitAtom}(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle))\\
=\ &\mathbf{splitAtom}(\delta(\langle C_1\rangle) \oplus_p \delta(\langle C_2\rangle))\\
=\ &\mathbf{splitAtom}(\delta(\langle C_1\rangle)) \oplus_p \mathbf{splitAtom}(\delta(\langle C_2\rangle)) \qquad \text{(by Lem. 122)}\\
=\ &\mathbf{splitAtom}(\langle C_1\rangle) \oplus_p \mathbf{splitAtom}(\langle C_2\rangle) \qquad \text{(by Lem. 118)}\\
=\ &\delta(\langle C_1\rangle) \oplus_p \delta(\langle C_2\rangle)\\
=\ &\mathbf{splitAtom}(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle)\\
=\ &\mathbf{splitAtom}(C).
\end{aligned}$$

– case 2: $C = \langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle\ sp$.

$$\begin{aligned}
&\mathbf{splitAtom}(\mathbf{splitAtom}(C))\\
=\ &\mathbf{splitAtom}(\mathbf{splitAtom}(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle\ sp))\\
=\ &\mathbf{splitAtom}(\delta(\langle C_1\rangle\ sp) \oplus_p \delta(\langle C_2\rangle\ sp))\\
=\ &\mathbf{splitAtom}(\delta(\langle C_1\rangle\ sp)) \oplus_p \mathbf{splitAtom}(\delta(\langle C_2\rangle\ sp)) \qquad \text{(by Lem. 122)}\\
=\ &\mathbf{splitAtom}(\langle C_1\rangle\ sp) \oplus_p \mathbf{splitAtom}(\langle C_2\rangle\ sp) \qquad \text{(by Lem. 118)}\\
=\ &\delta(\langle C_1\rangle\ sp) \oplus_p \delta(\langle C_2\rangle\ sp)\\
=\ &\mathbf{splitAtom}(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle\ sp)\\
=\ &\mathbf{splitAtom}(C).
\end{aligned}$$

– case 3: $C = C_1; C_2$.
IH: $\mathbf{splitAtom}(\mathbf{splitAtom}(C_1)) = \mathbf{splitAtom}(C_1)$.

$$\begin{aligned}
&\mathbf{splitAtom}(\mathbf{splitAtom}(C))\\
=\ &\mathbf{splitAtom}(\mathbf{splitAtom}(C_1; C_2))\\
=\ &\mathbf{splitAtom}(\mathbf{splitAtom}(C_1); C_2)\\
=\ &\mathbf{splitAtom}(\mathbf{splitAtom}(C_1)); C_2\\
=\ &\mathbf{splitAtom}(C_1); C_2 \qquad \text{(by IH)}\\
=\ &\mathbf{splitAtom}(C_1; C_2)\\
=\ &\mathbf{splitAtom}(C).
\end{aligned}$$

– other cases.
We have $\mathbf{splitAtom}(C) = \delta(C)$, thus

$$\begin{aligned}
&\mathbf{splitAtom}(\mathbf{splitAtom}(C))\\
=\ &\mathbf{splitAtom}(\delta(C))\\
=\ &\mathbf{splitAtom}(C). \qquad \text{(by Lem. 118)}
\end{aligned}$$

Lemma 134. *For all σ, σ', p, n, $(\textbf{\textit{skip}}, \sigma) \xrightarrow{p}{}^{n} (\textbf{\textit{skip}}, \sigma')$ if and only if $\sigma' = \sigma \wedge p = 1$ or $\sigma' \neq \sigma \wedge p = 0$.*

Proof. For all σ, σ', p, n, we prove by induction on n.

– base case: $n = 0$. trivial.

– inductive case: $n = k + 1$.
IH: $(\mathbf{skip}, \sigma) \xrightarrow{p}{}^{k} (\mathbf{skip}, \sigma')$ if and only if $\sigma' = \sigma \wedge p = 1$ or $\sigma' \neq \sigma \wedge p = 0$.

$$
\begin{array}{ll}
 & (\mathbf{skip}, \sigma) \xrightarrow{p}{}^{n} (\mathbf{skip}, \sigma') \\
\iff & (\mathbf{skip}, \sigma) \xrightarrow{p}{}^{k+1} (\mathbf{skip}, \sigma') \\
\iff & p = \sum_{C'', \sigma''} . \{p_1 \cdot p_2 \mid (\mathbf{skip}, \sigma) \xrightarrow{p_1} (C'', \sigma'') \wedge (C'', \sigma'') \xrightarrow{p_2}{}^{k} (\mathbf{skip}, \sigma')\} \\
\iff & (\mathbf{skip}, \sigma) \xrightarrow{p_2}{}^{k} (\mathbf{skip}, \sigma') \\
\iff & (\sigma' = \sigma \wedge p = 1) \vee (\sigma' \neq \sigma \wedge p = 0). \qquad \text{(by IH)}
\end{array}
$$

Lemma 135. *For all C, σ, σ', p, n such that $n \geq 1$, $(\langle C \rangle, \sigma) \xrightarrow{p}{}^{n} (\boldsymbol{skip}, \sigma')$ if and only if $(\langle C \rangle, \sigma) \xrightarrow{p} (\boldsymbol{skip}, \sigma')$.*

Proof. For all C, σ, σ', p, n such that $n \geq 1$, we have

$$
\begin{array}{ll}
 & (\langle C \rangle, \sigma) \xrightarrow{p}{}^{n} (\mathbf{skip}, \sigma') \\
\iff & p = \sum_{C'', \sigma''} \{p_1 \cdot p_2 \mid (\langle C \rangle, \sigma) \xrightarrow{p_1} (C'', \sigma'') \wedge (C'', \sigma'') \xrightarrow{p_2}{}^{n-1} (\mathbf{skip}, \sigma')\} \\
\iff & p = \sum_{\sigma''} \{p_1 \cdot p_2 \mid (\langle C \rangle, \sigma) \xrightarrow{p_1} (\mathbf{skip}, \sigma'') \wedge (\mathbf{skip}, \sigma'') \xrightarrow{p_2}{}^{n-1} (\mathbf{skip}, \sigma')\} \\
\iff & p = \sum_{\sigma''} \{p_1 \cdot p_2 \mid (\langle C \rangle, \sigma) \xrightarrow{p_1} (\mathbf{skip}, \sigma'') \wedge p_2 = 1 \wedge \sigma'' = \sigma'\} \qquad \text{(by Lem. 134)} \\
\iff & (\langle C \rangle, \sigma) \xrightarrow{p} (\mathbf{skip}, \sigma').
\end{array}
$$

Lemma 136. *For all C, σ, σ', p, $(\langle C \rangle, \sigma) \xrightarrow{p} (\boldsymbol{skip}, \sigma')$ if and only if there exists k such that $(\langle C \rangle, \sigma) \xrightarrow{p}{}^{n} (\boldsymbol{skip}, \sigma')$ for all $n \geq k$.*

Proof. For all C, σ, σ', p, we prove the two directions respectively.

– $(\langle C \rangle, \sigma) \xrightarrow{p} (\mathbf{skip}, \sigma')$.
Let $k \stackrel{\text{def}}{=} 1$. For all $n \geq k$, we know $n \geq 1$, by Lem. 135 we have $(\langle C \rangle, \sigma) \xrightarrow{p} {}^{n}(\mathbf{skip}, \sigma')$.

– there exists k such that $(\langle C \rangle, \sigma) \xrightarrow{p}{}^{n} (\mathbf{skip}, \sigma')$ for all $n \geq k$.
We know $(\langle C \rangle, \sigma) \xrightarrow{p}{}^{k+1} (\mathbf{skip}, \sigma')$. From $k + 1 \geq 1$ by Lem. 135 we have $(\langle C \rangle, \sigma) \xrightarrow{p} (\mathbf{skip}, \sigma')$.

Lemma 137. *For all C_1, C_2, σ, $\boldsymbol{step}(\langle \langle C_1 \rangle \oplus_p \langle C_2 \rangle \rangle, \sigma) = \boldsymbol{step}(\langle C_1 \rangle, \sigma) \oplus_p \boldsymbol{step}(\langle C_2 \rangle, \sigma)$.*

Proof. For all C_1, C_2, σ, we have

$$
\begin{aligned}
&\mathbf{step}(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle, \sigma)\\
&= \lambda(C',\sigma').\begin{cases} p', & \text{if } (\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle, \sigma) \xrightarrow{p'} (C',\sigma')\\ 0, & \text{otherwise}\end{cases}\\
&= \lambda(C',\sigma').\begin{cases} p', & \text{if } C' = \mathbf{skip} \wedge \exists k.\forall n \geq k.(\langle C_1\rangle \oplus_p \langle C_2\rangle, \sigma) \xrightarrow{p'}{}^{n} (\mathbf{skip},\sigma')\\ 0, & \text{otherwise}\end{cases}\\
&= \lambda(C',\sigma').\begin{cases} p\cdot p_1 + (1-p)\cdot p_2, & \text{if } C' = \mathbf{skip} \wedge \exists k \geq 1.\forall n \geq k.(\langle C_1\rangle, \sigma) \xrightarrow{p_1}{}^{n-1} (\mathbf{skip},\sigma') \wedge\\ & \qquad\qquad (\langle C_2\rangle, \sigma) \xrightarrow{p_2}{}^{n-1} (\mathbf{skip},\sigma')\\ 0, & \text{otherwise}\end{cases}\\
&= \lambda(C',\sigma').\begin{cases} p\cdot p_1 + (1-p)\cdot p_2, & \text{if } C' = \mathbf{skip} \wedge \exists k \geq 1.\forall n \geq k-1.(\langle C_1\rangle, \sigma) \xrightarrow{p_1}{}^{n} (\mathbf{skip},\sigma') \wedge\\ & \qquad\qquad (\langle C_2\rangle, \sigma) \xrightarrow{p_2}{}^{n} (\mathbf{skip},\sigma')\\ 0, & \text{otherwise}\end{cases}\\
&= \lambda(C',\sigma').\begin{cases} p\cdot p_1 + (1-p)\cdot p_2, & \text{if } C' = \mathbf{skip} \wedge \exists k.\forall n \geq k.(\langle C_1\rangle, \sigma) \xrightarrow{p_1}{}^{n} (\mathbf{skip},\sigma') \wedge\\ & \qquad\qquad (\langle C_2\rangle, \sigma) \xrightarrow{p_2}{}^{n} (\mathbf{skip},\sigma')\\ 0, & \text{otherwise}\end{cases}\\
&= \lambda(C',\sigma').\begin{cases} p\cdot p_1 + (1-p)\cdot p_2, & \text{if } C' = \mathbf{skip} \wedge (\exists k.\forall n \geq k.(\langle C_1\rangle, \sigma) \xrightarrow{p_1}{}^{n} (\mathbf{skip},\sigma')) \wedge\\ & \qquad\qquad (\exists k.\forall k.(\langle C_2\rangle, \sigma) \xrightarrow{p_2}{}^{n} (\mathbf{skip},\sigma'))\\ 0, & \text{otherwise}\end{cases}\\
&= \lambda(C',\sigma').\ p\cdot\begin{cases} p_1, & \text{if } C' = \mathbf{skip} \wedge (\exists k.\forall n \geq k.(\langle C_1\rangle, \sigma) \xrightarrow{p_1}{}^{n} (\mathbf{skip},\sigma'))\\ 0, & \text{otherwise}\end{cases}\\
&\qquad +(1-p)\cdot\begin{cases} p_2, & \text{if } C' = \mathbf{skip} \wedge (\exists k.\forall n \geq k.(\langle C_2\rangle, \sigma) \xrightarrow{p_2}{}^{n} (\mathbf{skip},\sigma'))\\ 0, & \text{otherwise}\end{cases}\\
&= \lambda(C',\sigma').\ p\cdot\begin{cases} p_1, & \text{if } (\langle C_1\rangle, \sigma) \xrightarrow{p_1} (C',\sigma')\\ 0, & \text{otherwise}\end{cases} + (1-p)\cdot\begin{cases} p_2, & \text{if } (\langle C_2\rangle, \sigma) \xrightarrow{p_2} (C',\sigma')\\ 0, & \text{otherwise}\end{cases} \qquad \text{(by Lem. 136)}\\
&= \lambda(C',\sigma').\ p\cdot\mathbf{step}(\langle C_1\rangle,\sigma)(C',\sigma') + (1-p)\cdot\mathbf{step}(\langle C_2\rangle,\sigma)(C',\sigma')\\
&= \mathbf{step}(\langle C_1\rangle,\sigma) \oplus_p \mathbf{step}(\langle C_2\rangle,\sigma).
\end{aligned}
$$

Lemma 138. *For all σ, σ, sp, $\boldsymbol{step}(\langle C\rangle\ sp, \sigma) = \boldsymbol{step}(\langle C\rangle, \sigma)$.*

Proof. For all σ, σ, sp, we have

$$
\begin{aligned}
&\mathbf{step}(\langle C\rangle\ sp, \sigma)\\
&= \lambda(C',\sigma').\begin{cases} p, & \text{if } (\langle C\rangle\ sp, \sigma) \xrightarrow{p} (C',\sigma')\\ 0, & \text{otherwise}\end{cases}\\
&= \lambda(C',\sigma').\begin{cases} p, & \text{if } C' = \mathbf{skip} \wedge (\langle C\rangle, \sigma) \xrightarrow{p} (\mathbf{skip},\sigma')\\ 0, & \text{otherwise}\end{cases}\\
&= \lambda(C',\sigma').\begin{cases} p, & \text{if } (\langle C\rangle, \sigma) \xrightarrow{p} (C',\sigma')\\ 0, & \text{otherwise}\end{cases}\\
&= \mathbf{step}(\langle C\rangle, \sigma).
\end{aligned}
$$

Lemma 139. *For all η and C_2, if $\eta^{(Stmt)}(\boldsymbol{skip}) = 0$, then $\boldsymbol{step}(\eta; C_2) = \boldsymbol{step}(\eta); C_2$.*

Proof. For all η and C_2 such that $\eta^{(Stmt)}(\mathbf{skip}) = 0$, we know $\sum_\sigma \eta(\mathbf{skip}, \sigma) = 0$, so $\eta(\mathbf{skip}, \sigma) = 0$ for all σ, thus

$$
\begin{aligned}
&\mathbf{step}(\eta; C_2)\\
&= \mathbb{E}_{(C,\sigma)\sim\eta;C_2}\{\mathbf{step}(C,\sigma)\}\\
&= \lambda(C',\sigma').\textstyle\sum_{C,\sigma}(\eta;C_2)(C,\sigma)\cdot\mathbf{step}(C,\sigma)(C',\sigma')\\
&= \lambda(C',\sigma').\textstyle\sum_{C_1,\sigma}\eta(C_1,\sigma)\cdot\mathbf{step}(C_1;C_2,\sigma)(C',\sigma')\\
&= \lambda(C',\sigma').\textstyle\sum_{C_1,\sigma}\{\eta(C_1,\sigma)\cdot\mathbf{step}(C_1;C_2,\sigma)(C',\sigma') \mid C_1 \neq \mathbf{skip}\}\\
&= \lambda(C',\sigma').\begin{cases}\sum_{C_1,\sigma}\eta(C_1,\sigma)\cdot\mathbf{step}(C_1,\sigma)(C_1',\sigma), & \text{if } C' = C_1';C_2\\ 0, \text{otherwise}\end{cases} && \text{(by Lem. 125)}\\
&= \lambda(C',\sigma').\begin{cases}\mathbb{E}_{(C_1,\sigma)\sim\eta}\{\mathbf{step}(C_1,\sigma)\}(C_1',\sigma), & \text{if } C' = C_1';C_2\\ 0, \text{otherwise}\end{cases}\\
&= \lambda(C',\sigma').\begin{cases}\mathbf{step}(\eta)(C_1',\sigma), & \text{if } C' = C_1';C_2\\ 0, \text{otherwise}\end{cases}\\
&= \mathbf{step}(\eta); C_2.
\end{aligned}
$$

Lemma 140. *For all C, σ, $\boldsymbol{step}(\mathbf{splitAtom}(C) \otimes \delta(\sigma)) = \boldsymbol{step}(C, \sigma)$.*

Proof. by induction on C.

- case 1: $C = \langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle$.
  For all σ, we have

$$
\begin{aligned}
&\mathbf{step}(\mathbf{splitAtom}(C) \otimes \delta(\sigma))\\
&= \mathbf{step}(\mathbf{splitAtom}(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle) \otimes \delta(\sigma))\\
&= \mathbf{step}((\delta(\langle C_1\rangle) \oplus_p \delta(\langle C_2\rangle)) \otimes \delta(\sigma))\\
&= \mathbf{step}((\delta(\langle C_1\rangle) \otimes \delta(\sigma)) \oplus_p (\delta(\langle C_2\rangle) \otimes \delta(\sigma))) && \text{(by Lem. 14)}\\
&= \mathbf{step}(\delta(\langle C_1\rangle) \otimes \delta(\sigma)) \oplus_p \mathbf{step}(\delta(\langle C_2\rangle) \otimes \delta(\sigma)) && \text{(by Lem. 124)}\\
&= \mathbf{step}(\langle C_1\rangle, \sigma) \oplus_p \mathbf{step}(\langle C_2\rangle, \sigma) && \text{(by Lem. 120)}\\
&= \mathbf{step}(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle, \sigma) && \text{(by Lem. 137)}\\
&= \mathbf{step}(C, \sigma).
\end{aligned}
$$

- case 2: $C = \langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle\ sp$.
  For all σ, we have

$$
\begin{aligned}
&\mathbf{step}(\mathbf{splitAtom}(C) \otimes \delta(\sigma))\\
&= \mathbf{step}(\mathbf{splitAtom}(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle\ sp) \otimes \delta(\sigma))\\
&= \mathbf{step}((\delta(\langle C_1\rangle\ sp) \oplus_p \delta(\langle C_2\rangle\ sp)) \otimes \delta(\sigma))\\
&= \mathbf{step}((\delta(\langle C_1\rangle\ sp) \otimes \delta(\sigma)) \oplus_p (\delta(\langle C_2\rangle\ sp) \otimes \delta(\sigma))) && \text{(by Lem. 14)}\\
&= \mathbf{step}(\delta(\langle C_1\rangle\ sp) \otimes \delta(\sigma)) \oplus_p \mathbf{step}(\delta(\langle C_2\rangle\ sp) \otimes \delta(\sigma)) && \text{(by Lem. 124)}\\
&= \mathbf{step}(\langle C_1\rangle\ sp, \sigma) \oplus_p \mathbf{step}(\langle C_2\rangle\ sp, \sigma) && \text{(by Lem. 120)}\\
&= \mathbf{step}(\langle C_1\rangle, \sigma) \oplus_p \mathbf{step}(\langle C_2\rangle, \sigma) && \text{(by Lem. 138)}\\
&= \mathbf{step}(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle, \sigma) && \text{(by Lem. 137)}\\
&= \mathbf{step}(\langle\langle C_1\rangle \oplus_p \langle C_2\rangle\rangle\ sp, \sigma) && \text{(by Lem. 138)}\\
&= \mathbf{step}(C, \sigma).
\end{aligned}
$$

– case 3: $C = C_1; C_2$.
IH: for all σ, $\mathbf{step}(\mathbf{splitAtom}(C_1) \otimes \delta(\sigma)) = \mathbf{step}(C_1, \sigma)$.
It is obvious that $C_1 = \mathbf{skip}$ or $C_1 \neq \mathbf{skip}$, we prove the two cases respectively.

- $C_1 = \mathbf{skip}$.
For all σ, we have

$$\begin{aligned}
&\mathbf{step}(\mathbf{splitAtom}(C) \otimes \delta(\sigma)) \\
=\ &\mathbf{step}(\mathbf{splitAtom}(\mathbf{skip}; C_2) \otimes \delta(\sigma)) \\
=\ &\mathbf{step}((\mathbf{splitAtom}(\mathbf{skip}); C_2) \otimes \delta(\sigma)) \\
=\ &\mathbf{step}((\delta(\mathbf{skip}); C_2) \otimes \delta(\sigma)) \\
=\ &\mathbf{step}(\delta(\mathbf{skip}; C_2) \otimes \delta(\sigma)) \\
=\ &\mathbf{step}(\mathbf{skip}; C_2, \sigma) \qquad \text{(by Lem. 120)} \\
=\ &\mathbf{step}(C, \sigma).
\end{aligned}$$

- $C_1 \neq \mathbf{skip}$.
For all σ, from $C_1 \neq \mathbf{skip}$ by Lem. 125 we know $\mathbf{step}(C_1; C_2, \sigma) = \mathbf{step}(C_1, \sigma); C_2$.
By Lem. 152 we know $\mathbf{splitAtom}(C_1) \otimes \delta(\sigma)^{(Stmt)}(\mathbf{skip}) = \mathbf{splitAtom}(C_1)(\mathbf{skip}) = \delta(C_1)(\mathbf{skip}) = 0$, by Lem. 139 we know $\mathbf{step}((\mathbf{splitAtom}(C_1) \otimes \delta(\sigma)); C_2) = \mathbf{step}(\mathbf{splitAtom}(C_1) \otimes \delta(\sigma)); C_2$, thus

$$\begin{aligned}
&\mathbf{step}(\mathbf{splitAtom}(C) \otimes \delta(\sigma)) \\
=\ &\mathbf{step}(\mathbf{splitAtom}(C_1; C_2) \otimes \delta(\sigma)) \\
=\ &\mathbf{step}((\mathbf{splitAtom}(C_1); C_2) \otimes \delta(\sigma)) \\
=\ &\mathbf{step}((\mathbf{splitAtom}(C_1) \otimes \delta(\sigma)); C_2) \qquad \text{(by Lem. 93)} \\
=\ &\mathbf{step}(\mathbf{splitAtom}(C_1) \otimes \delta(\sigma)); C_2 \\
=\ &\mathbf{step}(C_1, \sigma); C_2 \qquad \text{(by IH)} \\
=\ &\mathbf{step}(C_1; C_2, \sigma) \\
=\ &\mathbf{step}(C, \sigma).
\end{aligned}$$

– other cases.
We have $\mathbf{splitAtom}(C) = \delta(C)$, thus

$$\begin{aligned}
&\mathbf{step}(\mathbf{splitAtom}(C) \otimes \delta(\sigma)) \\
=\ &\mathbf{step}(\delta(C) \otimes \delta(\sigma)) \\
=\ &\mathbf{step}(C, \sigma). \qquad \text{(by Lem.120)}
\end{aligned}$$

Lemma 141. *For all* $\mathbb{C}, \sigma, t$, $\mathbf{splitAtom}(\boldsymbol{step}(\mathbf{splitAtom}(\mathbb{C}) \otimes \delta(\sigma), t)) = \mathbf{splitAtom}(\boldsymbol{step}(\mathbb{C}, \sigma, t))$.

Proof. For all $\mathbb{C}, \sigma, t$, by definition of *Prog* there exists $C_1, \ldots, C_n$ such that $\mathbb{C} = C_1 \parallel \cdots \parallel C_n$.

$$
\begin{aligned}
&\mathbf{splitAtom}(\mathbf{step}(\mathbf{splitAtom}(\mathbb{C}) \otimes \delta(\sigma), t)) \\
&= \mathbf{splitAtom}(\mathbf{step}(\mathbf{splitAtom}(C_1 \parallel \cdots \parallel C_n) \otimes \delta(\sigma), t)) \\
&= \mathbf{splitAtom}(\mathbf{step}(\mathbf{splitAtom}(C_1) \parallel \cdots \parallel \mathbf{splitAtom}(C_n) \otimes \delta(\sigma), t)) \\
&= \mathbf{splitAtom}(\mathbf{step}(\mathbf{splitAtom}(C_1) \parallel \cdots \parallel \mathbf{splitAtom}(C_{t-1}) \parallel \mathbf{splitAtom}(C_t) \otimes \delta(\sigma) \parallel \\
&\qquad \mathbf{splitAtom}(C_{t+1}) \parallel \cdots \parallel \mathbf{splitAtom}(C_n) \otimes \delta(\sigma), t)) \qquad \text{(by Lem. 131)} \\
&= \mathbf{splitAtom}(\mathbf{splitAtom}(C_1) \parallel \cdots \parallel \mathbf{splitAtom}(C_{t-1}) \parallel \mathbf{step}(\mathbf{splitAtom}(C_t) \otimes \delta(\sigma)) \parallel \\
&\qquad \mathbf{splitAtom}(C_{t+1}) \parallel \cdots \parallel \mathbf{splitAtom}(C_n)) \qquad \text{(by Lem. 132)} \\
&= \mathbf{splitAtom}(\mathbf{splitAtom}(C_1)) \parallel \cdots \parallel \mathbf{splitAtom}(\mathbf{splitAtom}(C_{t-1})) \parallel \mathbf{splitAtom}(\mathbf{step}(\mathbf{splitAtom}(C_t) \otimes \delta(\sigma))) \parallel \\
&\quad \mathbf{splitAtom}(\mathbf{splitAtom}(C_{t+1})) \parallel \cdots \parallel \mathbf{splitAtom}(\mathbf{splitAtom}(C_n)) \qquad \text{(by Lem. 114)} \\
&= \mathbf{splitAtom}(C_1) \parallel \cdots \parallel \mathbf{splitAtom}(C_{t-1}) \parallel \mathbf{splitAtom}(\mathbf{step}(C_t, \sigma)) \parallel \\
&\quad \mathbf{splitAtom}(C_{t+1}) \parallel \cdots \parallel \mathbf{splitAtom}(C_n) \qquad \text{(by Lem. 133 and Lem. 140)} \\
&= \mathbf{splitAtom}(C_1 \parallel \cdots \parallel C_{t-1} \parallel \mathbf{step}(C_t, \sigma) \parallel C_{t+1} \parallel \cdots \parallel C_n) \qquad \text{(by Lem. 114)} \\
&= \mathbf{splitAtom}(\mathbf{step}(C_1 \parallel \ldots C_n, \sigma, t)) \\
&= \mathbf{splitAtom}(\mathbf{step}(\mathbb{C}, \sigma, t)).
\end{aligned}
$$

Lemma 142. *For all W and t,* $\mathbf{splitAtom}(\boldsymbol{step}(\mathbf{splitAtom}(W), t)) = \mathbf{splitAtom}(\boldsymbol{step}(W, t))$.

Proof. For all W and t, we have

$$
\begin{aligned}
&\mathbf{splitAtom}(\mathbf{step}(\mathbf{splitAtom}(W), t)) \\
&= \mathbf{splitAtom}(\mathbf{step}(\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{splitAtom}(\mathbb{C}) \otimes \delta(\sigma)\}, t)) \\
&= \mathbf{splitAtom}(\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{step}(\mathbf{splitAtom}(\mathbb{C}) \otimes \delta(\sigma), t)\}) \qquad \text{(by Lem. 111)} \\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{splitAtom}(\mathbf{step}(\mathbf{splitAtom}(\mathbb{C}) \otimes \delta(\sigma), t))\} \qquad \text{(by Lem. 112)} \\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{splitAtom}(\mathbf{step}(\mathbb{C}, \sigma, t))\} \qquad \text{(by Lem. 141)} \\
&= \mathbf{splitAtom}(\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{step}(\mathbb{C}, \sigma, t)\}) \qquad \text{(by Lem. 112)} \\
&= \mathbf{splitAtom}(\mathbf{step}(W, t)).
\end{aligned}
$$

Lemma 143. *For all W_1, W_2, W_1', if $W_1 \sim W_2$ and $W_1 \overset{t}{\rightsquigarrow} W_1'$, then there exists W_2' such that $W_2 \overset{t}{\rightsquigarrow} W_2'$ and $W_1' \sim W_2'$.*

Proof. For all W_1, W_2, W_1' such that $W_1 \sim W_2$ and $W_1 \overset{t}{\rightsquigarrow} W_1'$, from $W_1 \sim W_2$ we know
$\mathbf{lazycoin}(W_1) = \mathbf{splitAtom}(W_2)$. From $W_1 \overset{t}{\rightsquigarrow} W_1'$ by Lem. 110 we know $W_1' = \mathbf{step}(W, t)$. Let $W_2' \overset{\text{def}}{=} \mathbf{step}(W_1, t)$, by Lem. 110 we know $W_2 \overset{t}{\rightsquigarrow} W_2'$.

$$
\begin{aligned}
&\mathbf{lazycoin}(W_1') \\
&= \mathbf{lazycoin}(\mathbf{step}(W_1, t)) \\
&= \mathbf{splitAtom}(\mathbf{step}(\mathbf{lazycoin}(W_1), t)) \qquad \text{(by Lem. 130)} \\
&= \mathbf{splitAtom}(\mathbf{step}(\mathbf{splitAtom}(W_2), t)) \\
&= \mathbf{splitAtom}(\mathbf{step}(W_2, t)) \qquad \text{(by Lem. 142)} \\
&= \mathbf{splitAtom}(W_2'),
\end{aligned}
$$

thus $W_1' \sim W_2'$.

Lemma 144. *For all W and b, if $[\![\boldsymbol{Pr}_b]\!]_{W^{(State)}} > 0$, then $\mathbf{lazycoin}(W|_b) = \mathbf{lazycoin}(W)|_b$.*

Proof. For all W and b such that $[\![\mathbf{Pr}_b]\!]_{W^{(State)}} > 0$, by Lem. 82 we know $W|_b = \lambda(\mathbb{C},\sigma).\frac{W(\mathbb{C},\sigma)\cdot\chi(\sigma\models b)}{[\![\mathbf{Pr}(b)]\!]_{W^{(State)}}}$, thus

$$
\begin{aligned}
&\mathbf{lazycoin}(W|_b)\\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim W|_b}\{\delta(\mathbf{lazycoin}(\mathbb{C}))\otimes\delta(\sigma)\}\\
&= \lambda(\mathbb{C}',\sigma').\textstyle\sum_{\mathbb{C},\sigma} W|_b(\mathbb{C},\sigma)\cdot\delta(\mathbf{lazycoin}(\mathbb{C}))(\mathbb{C}')\cdot\delta(\sigma)(\sigma')\\
&= \lambda(\mathbb{C}',\sigma').\textstyle\sum_{\mathbb{C},\sigma} W|_b(\mathbb{C},\sigma')\cdot\delta(\mathbf{lazycoin}(\mathbb{C}))(\mathbb{C}')\cdot\delta(\sigma)(\sigma')\\
&= \lambda(\mathbb{C}',\sigma').\textstyle\sum_{\mathbb{C}} \frac{W(\mathbb{C},\sigma')\cdot\chi(\sigma'\models b)}{[\![\mathbf{Pr}(b)]\!]_{W^{(State)}}}\cdot\delta(\mathbf{lazycoin}(\mathbb{C}))(\mathbb{C}')\cdot\delta(\sigma)(\sigma')\\
&= \lambda(\mathbb{C}',\sigma').\ \frac{\chi(\sigma'\models b)\cdot\sum_{\mathbb{C},\sigma} W(\mathbb{C},\sigma)\cdot\delta(\mathbf{lazycoin}(\mathbb{C}))(\mathbb{C}')\cdot\delta(\sigma)(\sigma')}{[\![\mathbf{Pr}(b)]\!]_{W^{(State)}}}\\
&= \lambda(\mathbb{C}',\sigma').\ \frac{\chi(\sigma'\models b)\cdot\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\delta(\mathbf{lazycoin}(\mathbb{C}))\otimes\delta(\sigma)\}(\mathbb{C}',\sigma')}{[\![\mathbf{Pr}(b)]\!]_{W^{(State)}}}\\
&= \lambda(\mathbb{C}',\sigma').\ \frac{\chi(\sigma'\models b)\cdot\mathbf{lazycoin}(W)(\mathbb{C}',\sigma')}{[\![\mathbf{Pr}(b)]\!]_{W^{(State)}}}\\
&= \lambda(\mathbb{C}',\sigma').\ \frac{\chi(\sigma'\models b)\cdot\mathbf{lazycoin}(W)(\mathbb{C}',\sigma')}{[\![\mathbf{Pr}(b)]\!]_{\mathbf{lazycoin}(W)^{(State)}}} \qquad \text{(by Lem. 106)}\\
&= \mathbf{lazycoin}(W)|_b. \qquad \text{(by Lem. 82)}
\end{aligned}
$$

Lemma 145. *For all W and b, if $[\![\boldsymbol{Pr}_b]\!]_{W^{(State)}} > 0$, then $\mathbf{splitAtom}(W|_b) = \mathbf{splitAtom}(W)|_b$.*

Proof. For all W and b such that $[\![\mathbf{Pr}_b]\!]_{W^{(State)}} > 0$, by Lem. 82 we know $W|_b = \lambda(\mathbb{C},\sigma).\frac{W(\mathbb{C},\sigma)\cdot\chi(\sigma\models b)}{[\![\mathbf{Pr}(b)]\!]_{W^{(State)}}}$, thus

$$
\begin{aligned}
&\mathbf{splitAtom}(W|_b)\\
&= \mathbb{E}_{(\mathbb{C},\sigma)\sim W|_b}\{\mathbf{splitAtom}(\mathbb{C})\otimes\delta(\sigma)\}\\
&= \lambda(\mathbb{C}',\sigma').\textstyle\sum_{\mathbb{C},\sigma} W|_b(\mathbb{C},\sigma)\cdot\delta(\mathbf{splitAtom}(\mathbb{C}))(\mathbb{C}')\cdot\delta(\sigma)(\sigma')\\
&= \lambda(\mathbb{C}',\sigma').\textstyle\sum_{\mathbb{C},\sigma} W|_b(\mathbb{C},\sigma')\cdot\mathbf{splitAtom}(\mathbb{C})(\mathbb{C}')\cdot\delta(\sigma)(\sigma')\\
&= \lambda(\mathbb{C}',\sigma').\textstyle\sum_{\mathbb{C}} \frac{W(\mathbb{C},\sigma')\cdot\chi(\sigma'\models b)}{[\![\mathbf{Pr}(b)]\!]_{W^{(State)}}}\cdot\mathbf{splitAtom}(\mathbb{C})(\mathbb{C}')\cdot\delta(\sigma)(\sigma')\\
&= \lambda(\mathbb{C}',\sigma').\ \frac{\chi(\sigma'\models b)\cdot\sum_{\mathbb{C},\sigma} W(\mathbb{C},\sigma)\cdot\mathbf{splitAtom}(\mathbb{C})(\mathbb{C}')\cdot\delta(\sigma)(\sigma')}{[\![\mathbf{Pr}(b)]\!]_{W^{(State)}}}\\
&= \lambda(\mathbb{C}',\sigma').\ \frac{\chi(\sigma'\models b)\cdot\mathbb{E}_{(\mathbb{C},\sigma)\sim W}\{\mathbf{splitAtom}(\mathbb{C})\otimes\delta(\sigma)\}(\mathbb{C}',\sigma')}{[\![\mathbf{Pr}(b)]\!]_{W^{(State)}}}\\
&= \lambda(\mathbb{C}',\sigma').\ \frac{\chi(\sigma'\models b)\cdot\mathbf{splitAtom}(W)(\mathbb{C}',\sigma')}{[\![\mathbf{Pr}(b)]\!]_{W^{(State)}}}\\
&= \lambda(\mathbb{C}',\sigma').\ \frac{\chi(\sigma'\models b)\cdot\mathbf{splitAtom}(W)(\mathbb{C}',\sigma')}{[\![\mathbf{Pr}(b)]\!]_{\mathbf{splitAtom}(W)^{(State)}}} \qquad \text{(by Lem. 107)}\\
&= \mathbf{splitAtom}(W)|_b. \qquad \text{(by Lem. 82)}
\end{aligned}
$$

Lemma 146. *For all W_1, W_2, if $W_1 \sim W_2$ and $[\![\boldsymbol{Pr}_b]\!]_{W_1{}^{(State)}} > 0$, then $W_1|_b \sim W_2|_b$.*

Proof. For all W_1, W_2 such that $W_1 \sim W_2$ and $[\![\mathbf{Pr}_b]\!]_{W_1{}^{(State)}} > 0$, from $W_1 \sim W_2$ by Lem. 108 we know $W_1{}^{(State)} = W_2{}^{(State)}$, so $[\![\mathbf{Pr}_b]\!]_{W_2{}^{(State)}} = [\![\mathbf{Pr}_b]\!]_{W_1{}^{(State)}} > 0$. From $W_1 \sim W_2$ we know $\mathbf{lazycoin}(W_1) = \mathbf{splitAtom}(W_2)$. From $[\![\mathbf{Pr}_b]\!]_{W_1{}^{(State)}} > 0$ by Lem. 144 we know $\mathbf{lazycoin}(W_1|_b) = \mathbf{lazycoin}(W_1)|_b$. From $[\![\mathbf{Pr}_b]\!]_{W_2{}^{(State)}} > 0$ by Lem. 145 we know $\mathbf{splitAtom}(W_2|_b) = \mathbf{splitAtom}(W_2)|_b$, thus $\mathbf{lazycoin}(W_1|_b) = \mathbf{lazycoin}(W_1)|_b = \mathbf{splitAtom}(W_2)|_b = \mathbf{splitAtom}(W_2|_b)$, so $W_1|_b \sim W_2|_b$.

Lemma 147. *For all W_1, W_2, W_1', if $W_1 \sim W_2$ and $W_1 \overset{t}{\hookrightarrow} W_1'$, then there exists W_2' such that $W_2 \overset{t}{\hookrightarrow} W_2'$ and $W_1' \sim W_2'$.*

Proof. For all W_1, W_2, W_1' such that $W_1 \sim W_2$ and $W_1 \overset{t}{\hookrightarrow} W_1'$, there are two cases.

- $\mathit{nextsplit}(W_1) = \{\mathbf{split}(b_1, \ldots, b_k)\}$, $W_1 \overset{t}{\rightsquigarrow} W_1''$ and $W_1''|_{b_i} = W_1'$.
  From $W_1 \sim W_2$ by Lem.105 we know $\mathit{nextsplit}(W_2, t) = \mathit{nextsplit}(W_1, t) = \{\mathbf{split}(b_1, \ldots, b_k)\}$. From $W_1 \sim W_2$ and $W_1 \overset{t}{\rightsquigarrow} W_1''$ by Lem. 143 there exists W_2'' such that $W_2 \overset{t}{\rightsquigarrow} W_2''$ and $W_1'' \sim W_2''$. From $W_1''|_{b_i} = W_1'$ by Lem. 63 we know $[\![\mathbf{Pr}(b_i)]\!]_{W_1''(State)} > 0$. From $W_1'' \sim W_2''$ by Lem. 146 we know $W_1''|_{b_i} \sim W_2''|_{b_i}$. Let $W_2' \overset{\text{def}}{=} W_2''|_{b_i}$, then $W_1' \sim W_2'$. From $\mathit{nextsplit}(W_2, t) = \{\mathbf{split}(b_1, \ldots, b_k)\}$, $W_2 \overset{t}{\rightsquigarrow} W_2''$ and $W_2''|_{b_i} = W_2'$ we have $W_2 \overset{t}{\hookrightarrow} W_2'$.
- $\#\mathit{nextsplit}(W_1, t) = 1$ and $W_1 \overset{t}{\rightsquigarrow} W_1'$.
  From $W_1 \sim W_2$ by Lem.105 we know $\mathit{nextsplit}(W_2, t) = \mathit{nextsplit}(W_1, t)$, so $\#\mathit{nextsplit}(W_2, t) = \#\mathit{nextsplit}(W_1, t) = 1$. From $W_1 \sim W_2$ and $W_1 \overset{t}{\rightsquigarrow} W_1'$ by Lem. 143 there exists W_2' such that $W_2 \overset{t}{\rightsquigarrow} W_2'$ and $W_1' \sim W_2'$. From $\#\mathit{nextsplit}(W_2, t) = 1$ we know $W_2 \overset{t}{\hookrightarrow} W_2'$.

Lemma 148. *For all $W_1, W_2, \varphi, \vec{W}_1, \vec{W}_2$, if $W_1 \sim W_2$ and $\boldsymbol{History}(W_1, \varphi, \vec{W}_1)$, then there exists $\vec{W}_2$ such that $\boldsymbol{History}(W_2, \varphi, \vec{W}_2)$ and $\vec{W}_1[n] \sim \vec{W}_2[n]$ for all n.*

Proof. by coinduction. For all $W_1, W_2, \varphi, \vec{W}_1, \vec{W}_2$ such that $W_1 \sim W_2$ and $\mathbf{History}(W_1, \varphi, \vec{W}_1)$, there exists $t, \varphi', W_1', \vec{W}_1'$ such that $\varphi = t :: \varphi'$, $W_1 \overset{t}{\hookrightarrow} W_1'$, $\mathbf{History}(W_1', \varphi', \vec{W}_1')$ and $\vec{W}_1 = W_1 :: \vec{W}_1'$. From $W_1 \sim W_2$ and $W_1 \overset{t}{\hookrightarrow} W_1'$ by Lem. 147 there exists W_2' such that $W_2 \overset{t}{\hookrightarrow} W_2'$ and $W_1' \sim W_2'$. From $\mathbf{History}(W_1', \varphi', \vec{W}_1')$ by coinduction hypothesis there exists $\vec{W}_2'$ such that $\mathbf{History}(W_2', \varphi', \vec{W}_2')$ and $\vec{W}_1'[n] \sim \vec{W}_2'[n]$ for all n. Let $\vec{W}_2 \overset{\text{def}}{=} W_2 :: \vec{W}_2'$, from $W_2 \overset{t}{\hookrightarrow} W_2'$ and $\mathbf{History}(W_2', \varphi', \vec{W}_2')$ we know $\mathbf{History}(W_2, t :: \varphi', W_2 :: \vec{W}_2')$, i.e., $\mathbf{History}(W_2, \varphi, \vec{W}_2)$. For all n, it is obvious that $n = 0$ or $n > 0$, we prove $\vec{W}_1[n] \sim \vec{W}_2$ in the two cases respectively.

- $n = 0$.
  $\vec{W}_1[0] = (W_1 :: \vec{W}_1')[0] = W_1$. $\vec{W}_2[0] = (W_2 :: \vec{W}_2')[0] = W_2$. From $W_1 \sim W_2$ we know $\vec{W}_1[0] \sim \vec{W}_2[0]$.

– $n > 0$.
$\vec{W}_1[n] = (W_1 :: \vec{W}_1')[n] = \vec{W}_1'[n-1]$. $\vec{W}_2[n] = (W_2 :: \vec{W}_2')[n] = \vec{W}_1'[n-1]$.
From $\vec{W}_1'[n-1] \sim \vec{W}_2'[n-1]$ we know $\vec{W}_1[n] \sim \vec{W}_2[n]$.

Lemma 149. *For all C,* $\mathbf{lazycoin}(C) = \boldsymbol{skip}$ *if and only if* $C = \boldsymbol{skip}$.

Proof. For all C, it is obvious that $C = \mathbf{skip}$ or $C \neq \mathbf{skip}$, we prove the two cases respectively.

– $C = \mathbf{skip}$.
Both $\mathbf{lazycoin}(C) = \mathbf{skip}$ and $C = \mathbf{skip}$ are true.
– $C \neq \mathbf{skip}$.
Both $\mathbf{lazycoin}(C) = \mathbf{skip}$ and $C = \mathbf{skip}$ are false.

Lemma 150. *For all $\mathbb{C}$,* $\mathbf{lazycoin}(\mathbb{C}) = \boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}$ *if and only if* $\mathbb{C} = \boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}$.

Proof. For all $\mathbb{C}$, there exists $C_1, \ldots, C_n$ such that $\mathbb{C} = C_1 \parallel \cdots \parallel C_n$.

$$
\begin{aligned}
& \mathbf{lazycoin}(\mathbb{C}) = \mathbf{skip} \parallel \cdots \parallel \mathbf{skip} \\
\iff & \mathbf{lazycoin}(C_1 \parallel \cdots \parallel C_n) = \mathbf{skip} \parallel \cdots \parallel \mathbf{skip} \\
\iff & \mathbf{lazycoin}(C_1) \parallel \cdots \parallel \mathbf{lazycoin}(C_n) = \mathbf{skip} \parallel \cdots \parallel \mathbf{skip} \\
\iff & \mathbf{lazycoin}(C_1) = \mathbf{skip} \wedge \cdots \wedge \mathbf{lazycoin}(C_n) = \mathbf{skip} \\
\iff & C_1 = \mathbf{skip} \wedge \cdots \wedge C_n = \mathbf{skip} \qquad \text{(by Lem. 149)} \\
\iff & C_1 \parallel \cdots \parallel C_n = \mathbf{skip} \parallel \cdots \parallel \mathbf{skip} \\
\iff & \mathbb{C} = \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}.
\end{aligned}
$$

Lemma 151. *For all W and σ,* $\mathbf{lazycoin}(W)(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}, \sigma) = W(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}, \sigma)$.

Proof. For all W and σ,

$$
\begin{aligned}
& \mathbf{lazycoin}(W)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \\
= & \mathbb{E}_{(\mathbb{C},\sigma_1)\sim W}\{\delta(\mathbf{lazycoin}(W)) \otimes \delta(\sigma_1)\}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \\
= & \textstyle\sum_{\mathbb{C},\sigma_1} W(\mathbb{C}, \sigma_1) \cdot \delta(\mathbf{lazycoin}(\mathbb{C}))(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \cdot \delta(\sigma_1)(\sigma) \\
= & \textstyle\sum_{\mathbb{C}} W(\mathbb{C}, \sigma) \cdot \delta(\mathbf{lazycoin}(\mathbb{C}))(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \\
= & \textstyle\sum_{\mathbb{C}} \{W(\mathbb{C}, \sigma) \mid \mathbf{lazycoin}(\mathbb{C}) = \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\} \\
= & \textstyle\sum_{\mathbb{C}} \{W(\mathbb{C}, \sigma) \mid \mathbb{C} = \mathbf{skip} \parallel \cdots \parallel \mathbf{skip}\} \qquad \text{(by Lem. 150)} \\
= & W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma).
\end{aligned}
$$

Lemma 152. *For all C,* $\mathbf{splitAtom}(C)(\boldsymbol{skip}) = \delta(C)(\boldsymbol{skip})$.

Proof. For all C, it is obvious that $C = \mathbf{skip}$ or $C \neq \mathbf{skip}$, we prove the two cases respectively.

– $C = \mathbf{skip}$.
$\mathbf{splitAtom}(C)(\mathbf{skip}) = \mathbf{splitAtom}(\mathbf{skip})(\mathbf{skip}) = \delta(\mathbf{skip})(\mathbf{skip}) = \delta(C)(\mathbf{skip})$.
– $C \neq \mathbf{skip}$.
$\mathbf{splitAtom}(C)(\mathbf{skip}) = 0 = \delta(C)(\mathbf{skip})$.

Lemma 153. *For all* $\mathbb{C}$, $\mathbf{splitAtom}(\mathbb{C})(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}) = \delta(\mathbb{C})(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip})$.

Proof. For all $\mathbb{C}$, there exists $C_1, \ldots, C_n$ such that $\mathbb{C} = C_1 \parallel \cdots \parallel C_n$.

$$
\begin{aligned}
&\mathbf{splitAtom}(\mathbb{C})(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \\
&= \mathbf{splitAtom}(C_1 \parallel \cdots \parallel C_n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \\
&= (\mathbf{splitAtom}(C_1) \parallel \cdots \parallel \mathbf{splitAtom}(C_n))(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \\
&= \mathbf{splitAtom}(C_1)(\mathbf{skip}) * \cdots * \mathbf{splitAtom}(C_n)(\mathbf{skip}) \\
&= \delta(C_1)(\mathbf{skip}) * \cdots * \delta(C_n)(\mathbf{skip}) \qquad \text{(by Lem. 152)} \\
&= (\delta(C_1) \parallel \cdots \parallel \delta(C_n))(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \\
&= \delta(C_1 \parallel \cdots \parallel C_n)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \qquad \text{(by Lem. 98)} \\
&= \delta(\mathbb{C})(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}).
\end{aligned}
$$

Lemma 154. *For all* W *and* σ, $\mathbf{splitAtom}(W)(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}, \sigma) = W(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}, \sigma)$.

Proof. For all W and σ,

$$
\begin{aligned}
&\mathbf{splitAtom}(W)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \\
&= \mathbb{E}_{(\mathbb{C},\sigma_1)\sim W}\{\mathbf{splitAtom}(W) \otimes \delta(\sigma_1)\}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \\
&= \textstyle\sum_{\mathbb{C},\sigma_1} W(\mathbb{C}, \sigma_1) \cdot \mathbf{splitAtom}(\mathbb{C})(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \cdot \delta(\sigma_1)(\sigma) \\
&= \textstyle\sum_{\mathbb{C}} W(\mathbb{C}, \sigma) \cdot \mathbf{splitAtom}(\mathbb{C})(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \\
&= \textstyle\sum_{\mathbb{C}} W(\mathbb{C}, \sigma) \cdot \delta(\mathbb{C})(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) \qquad \text{(by Lem. 153)} \\
&= W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma).
\end{aligned}
$$

Lemma 155. *For all* W_1 *and* W_2, *if* $W_1 \sim W_2$, *then* $W_1(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}, \sigma) = W_2(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}, \sigma)$ *for all* σ.

Proof. For all W_1 and W_2 such that $W_1 \sim W_2$, we know $\mathbf{lazycoin}(W_1) = \mathbf{splitAtom}(W_2)$. For all σ, we have

$$
\begin{aligned}
&W_1(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \\
&= \mathbf{lazycoin}(W_1)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \qquad \text{(by Lem. 151)} \\
&= \mathbf{splitAtom}(W_2)(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) \\
&= W_2(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma). \qquad \text{(by Lem. 154)}
\end{aligned}
$$

Lemma 156 (Soundness of (LAZYCOIN) Rule). *For all* $P, \mathbb{C}, Q$, *if* $\models_{A} \{P\}\mathbf{lazycoin}(\mathbb{C})\{Q\}$, *then* $\models_{A} \{P\}\mathbb{C}\{Q\}$.

Proof. For all $P, \mathbb{C}, Q$ such that $\models_{A} \{P\}\mathbf{lazycoin}(\mathbb{C})\{Q\}$, by Lem. 44 we have $\models_{A'} \{P\}\mathbf{lazycoin}(\mathbb{C})\{Q\}$ and we need to prove $\models_{A'} \{P\}\mathbb{C}\{Q\}$. By Def. 40, we need to prove for all μ, φ, μ', if $\mu \models P$ and $\mathit{init}(\mathbb{C}, \mu) \Downarrow'_{\varphi} \mu'$, then $\mu' \models Q$. For all μ, φ, μ' such that $\mu \models P$ and $\mathit{init}(\mathbb{C}, \mu) \Downarrow'_{\varphi} \mu'$. from $\mathit{init}(\mathbb{C}, \mu) \Downarrow'_{\varphi} \mu'$ we know there exists $\vec{W}$ such that $\mathbf{History}(\mathit{init}(\mathbb{C}, \mu), \varphi, \vec{W})$, $\lim_{n\to\infty} \vec{W}[n] \overset{(Prog)}{(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})} = 1$ and $\forall\sigma.\ \lim_{n\to\infty} \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) = \mu'(\sigma)$. By Lem. 100 we

know $\mathit{init}(\mathbb{C},\mu) \sim \mathit{init}(\mathbf{lazycoin}(\mathbb{C}),\mu)$. From $\mathbf{History}(W,\varphi,\vec{W})$ by Lem. 148 we know there exists $\vec{W}'$ such that $\mathbf{History}(\mathit{init}(\mathbf{lazycoin}(\mathbb{C}),\mu),\varphi,\vec{W}')$ and $\vec{W}[n] \sim \vec{W}'[n]$ for all n. For all n, from $\vec{W}[n] \sim \vec{W}'[n]$ by Lem. 155 we know $\vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip},\sigma) = \vec{W}'[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip},\sigma)$ for all σ, thus $\vec{W}[n] \overset{(Prog)}{(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})} = \sum_\sigma \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip},\sigma) = \sum_\sigma \vec{W}'[n] \overset{(Prog)}{(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip},\sigma)} = \vec{W}'[n] \overset{(Prog)}{(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})}$. Therefore $\lim_{n\to\infty} \vec{W}'[n] \overset{(Prog)}{(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})} = \lim_{n\to\infty} \vec{W}[n] \overset{(Prog)}{(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip})} = 1$ and $\lim_{n\to\infty} \vec{W}'[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip},\sigma) = \lim_{n\to\infty} \vec{W}[n](\mathbf{skip} \parallel \cdots \parallel \mathbf{skip},\sigma) = \mu'(\sigma)$ for all σ, so $\mathit{init}(\mathbf{lazycoin}(\mathbb{C}),\mu) \Downarrow'_\varphi \mu'$. From $\models_{\mathrm{A}'} \{P\}\mathbf{lazycoin}(\mathbb{C})\{Q\}$ and $\mu \models P$ we know $\mu' \models Q$.

Lemma 157 (Soundness of (P-CSQ) rule). *For all $P, P_1, \mathbb{C}, Q_1, Q$, if $P \Rightarrow P_1$, $\models_{\mathrm{A}} \{P_1\}\mathbb{C}\{Q_1\}$, and $Q_1 \Rightarrow Q$, then $\models_{\mathrm{A}} \{P\}\mathbb{C}\{Q\}$.*

Proof. For all $P, P_1, \mathbb{C}, Q_1, Q$ such that $P \Rightarrow P_1$, $\models_{\mathrm{A}} \{P_1\}\mathbb{C}\{Q_1\}$, and $Q_1 \Rightarrow Q$, to prove $\models_{\mathrm{A}} \{P\}\mathbb{C}\{Q\}$, we need to prove for all μ, φ, W, if $\mu \models P$ and $\mathit{init}(\mathbb{C},\mu) \Downarrow_\varphi W$, then $W^{(State)} \models Q$. For all μ, φ, W such that $\mu \models P$ and $\mathit{init}(\mathbb{C},\mu) \Downarrow_\varphi W$, from $\mu \models P$ and $P \Rightarrow P_1$ we know $\mu \models P_1$. From $\models_{\mathrm{A}} \{P_1\}\mathbb{C}\{Q_1\}$ and $\mathit{init}(\mathbb{C},\mu) \Downarrow_\varphi W$ we know $W^{(State)} \models Q_1$. From $Q_1 \Rightarrow Q$ we know $W^{(State)} \models Q$.

Lemma 158 (Soundness of (BIGCONJ) rule). *For all $\mathbb{C}, P_1, \ldots, P_n, Q_1, \ldots, Q_n$, if $\models_{\mathrm{A}} \{P_1\}\mathbb{C}\{Q_1\}, \ldots, \models_{\mathrm{A}} \{P_n\}\mathbb{C}\{Q_n\}$, then $\models_{\mathrm{A}} \{P_1 \wedge \cdots \wedge P_n\}\mathbb{C}\{Q_1 \wedge \cdots \wedge Q_n\}$.*

Proof. For all $\mathbb{C}, P_1, \ldots, P_n, Q_1, \ldots, Q_n$ such that $\models_{\mathrm{A}} \{P_1\}\mathbb{C}\{Q_1\}, \ldots, \models_{\mathrm{A}} \{P_n\}\mathbb{C}\{Q_n\}$, to prove $\models_{\mathrm{A}} \{P_1 \wedge \cdots \wedge P_n\}\mathbb{C}\{Q_1 \wedge \cdots \wedge Q_n\}$, we need to prove for all μ, φ, W, if $\mu \models P_1 \wedge \cdots \wedge P_n$ and $\mathit{init}(\mathbb{C},\mu) \Downarrow_\varphi W$, then $W^{(State)} \models Q_1 \wedge \cdots \wedge Q_n$. For all μ, φ, W such that $\mu \models P_1 \wedge \cdots \wedge P_n$ and $\mathit{init}(\mathbb{C},\mu) \Downarrow_\varphi W$, from $\mu \models P_1 \wedge \cdots \wedge P_n$ we know $\mu \models P_1$, …, $\mu \models P_n$. For all $i \in \{1, \ldots, n\}$, from $\models_{\mathrm{A}} \{P_i\}\mathbb{C}\{Q_i\}$, $\mu \models P_i$ and $\mathit{init}(\mathbb{C},\mu) \Downarrow_\varphi W$ we know $W^{(State)} \models Q_i$. Therefore $W^{(State)} \models Q_1 \wedge \cdots \wedge Q_n$.

Lemma 159 (Soundness of (BIGDISJ) rule). *For all $\mathbb{C}, P_1, \ldots, P_n, Q_1, \ldots, Q_n$, if $\models_{\mathrm{A}} \{P_1\}\mathbb{C}\{Q_1\}, \ldots, \models_{\mathrm{A}} \{P_n\}\mathbb{C}\{Q_n\}$, then $\models_{\mathrm{A}} \{P_1 \vee \cdots \vee P_n\}\mathbb{C}\{Q_1 \vee \cdots \vee Q_n\}$.*

Proof. For all $\mathbb{C}, P_1, \ldots, P_n, Q_1, \ldots, Q_n$ such that $\models_{\mathrm{A}} \{P_1\}\mathbb{C}\{Q_1\}, \ldots, \models_{\mathrm{A}} \{P_n\}\mathbb{C}\{Q_n\}$, to prove $\models_{\mathrm{A}} \{P_1 \vee \cdots \vee P_n\}\mathbb{C}\{Q_1 \vee \cdots \vee Q_n\}$, we need to prove for all μ, φ, W, if $\mu \models P_1 \vee \cdots \vee P_n$ and $\mathit{init}(\mathbb{C},\mu) \Downarrow_\varphi W$, then $W^{(State)} \models Q_1 \vee \cdots \vee Q_n$. For all μ, φ, W such that $\mu \models P_1 \vee \cdots \vee P_n$ and $\mathit{init}(\mathbb{C},\mu) \Downarrow_\varphi W$, from $\mu \models P_1 \vee \cdots \vee P_n$ we know there exists i such that $\mu \models P_i$. From $\models_{\mathrm{A}} \{P_i\}\mathbb{C}\{Q_i\}$, $\mu \models P_i$ and $\mathit{init}(\mathbb{C},\mu) \Downarrow_\varphi W$ we know $W^{(State)} \models Q_i$. Therefore $W^{(State)} \models Q_1 \vee \cdots \vee Q_n$.

Definition 73. $W \Longmapsto^n_\varphi (I, Q)$ is inductively defined as follows: $W \Longmapsto^0_\varphi (I, Q)$ always holds; $W \Longmapsto^{n+1}_{t::\varphi} (I, Q)$ holds if and and only if the following are true:

1. $W^{(State)} \models I$;
2. if $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$, then $W|_{\mathbf{skip}\parallel\cdots\parallel\mathbf{skip}}{}^{(State)} \models Q$;
3. for all W', if $W \overset{t}{\hookrightarrow} W'$, then $W' \Longmapsto^n_\varphi (I, Q)$.

Here $W|_{\mathbf{skip}\parallel\cdots\parallel\mathbf{skip}} = W|_{\lambda(\mathbb{C},\sigma).\mathbb{C}=\mathbf{skip}\parallel\cdots\parallel\mathbf{skip}}$.

Definition 74. $I \models_{ANL} \{P\}\mathbb{C}\{Q\}$ iff for all μ, if $\mu \models I \wedge P$, then $init(\mathbb{C}, \mu) \Longmapsto^n_\varphi (I, Q)$ for all φ and n.

Definition 75. Let $W \in \mathbb{D}_{Prog^n \times State}$, where $Prog^n$ means "programs with n threads". We define $\pi_i(W) \overset{\text{def}}{=} \lambda(C, \sigma').\ \mathbf{Pr}_{(C_1\parallel\cdots\parallel C_n,\sigma)\sim W}[C_i = C \wedge \sigma = \sigma']$.

Lemma 160. *For all $W \in \mathbb{D}_{Prog^n \times State}$ and i, $\pi_i(W) = \lambda(C_i, \sigma).\ \sum_{C_1,\ldots,C_{i-1},C_{i+1},\ldots,C_n} W(C_1 \parallel \cdots \parallel C_n, \sigma)$.*

Proof. For all $W \in \mathbb{D}_{Prog^n \times State}$ and i,

$$\begin{aligned}
\pi_i(W) &= \lambda(C, \sigma').\ \mathbf{Pr}_{(C_1\parallel\cdots\parallel C_n,\sigma)\sim W}[C_i = C \wedge \sigma = \sigma'] \\
&= \lambda(C, \sigma').\ \textstyle\sum_{C_1,\ldots,C_n,\sigma}\{W(C_1 \parallel \cdots \parallel C_n, \sigma) \mid C_i = C \wedge \sigma = \sigma'\} \\
&= \lambda(C, \sigma').\ \textstyle\sum_{C_1,\ldots,C_{i-1},C_{i+1},\ldots,C_n} W(C_1 \parallel \ldots C_{t-1} \parallel C \parallel C_{t+1} \parallel \cdots \parallel C_n, \sigma') \\
&= \lambda(C_i, \sigma).\ \textstyle\sum_{C_1,\ldots,C_{i-1},C_{i+1},\ldots,C_n} W(C_1 \parallel \cdots \parallel C_n, \sigma).
\end{aligned}$$

Lemma 161. *For all $C_1, \ldots, C_n, \mu, i$, $\pi_i(init(C_1 \parallel \cdots \parallel C_n, \mu)) = init(C_i, \mu)$ for all i.*

Proof. For all $C_1, \ldots, C_n, \mu, i$,

$$\begin{aligned}
&\pi_i(init(C_1 \parallel \cdots \parallel C_n, \mu)) \\
&= \pi_i(\delta(C_1 \parallel \cdots \parallel C_n) \otimes \mu) \\
&= \lambda(C'_i, \sigma).\ \textstyle\sum_{C'_1,\ldots,C'_{i-1},C'_{i+1},\ldots,C'_n} (\delta(C_1 \parallel \cdots \parallel C_n) \otimes \mu)(C'_1 \parallel \cdots \parallel C'_n, \sigma) && \text{(by Lem. 160)} \\
&= \lambda(C'_i, \sigma).\ \textstyle\sum_{C'_1,\ldots,C'_{i-1},C'_{i+1},\ldots,C'_n} \delta(C_1 \parallel \cdots \parallel C_n)(C'_1 \parallel \cdots \parallel C'_n) \cdot \mu(\sigma) \\
&= \lambda(C'_i, \sigma).\ \textstyle\sum_{C'_1,\ldots,C'_{i-1},C'_{i+1},\ldots,C'_n} (\delta(C_1) \parallel \cdots \parallel \delta(C_n))(C'_1 \parallel \cdots \parallel C'_n) \cdot \mu(\sigma) && \text{(by Lem. 98)} \\
&= \lambda(C'_i, \sigma).\ \textstyle\sum_{C'_1,\ldots,C'_{i-1},C'_{i+1},\ldots,C'_n} \delta(C_1)(C'_1) \cdots \delta(C_n)(C'_n) \cdot \mu(\sigma) \\
&= \lambda(C'_i, \sigma).\ \delta(C_i)(C'_i) \cdot \mu(\sigma) \\
&= init(C_i, \mu).
\end{aligned}$$

Lemma 162. *For all $W \in \mathbb{D}_{Prog^n \times State}$ and i, $\pi_i(W)^{(State)} = W^{(State)}$.*

Proof. For all $W \in \mathbb{D}_{Prog^n \times State}$ and i,

$$\begin{aligned}
\pi_i(W)^{(State)} &= \lambda\sigma.\ \textstyle\sum_{C_i} \pi_i(W)(C_i, \sigma) \\
&= \lambda\sigma.\ \textstyle\sum_{C_i} \sum_{C_1,\ldots,C_{t-1},C_{t+1},\ldots,C_n} W(C_1 \parallel \cdots \parallel C_n, \sigma) && \text{(by Lem. 160)} \\
&= \lambda\sigma.\ \textstyle\sum_{C_1,\ldots,C_n} W(C_1 \parallel \cdots \parallel C_n, \sigma) \\
&= \lambda\sigma.\ \textstyle\sum_{\mathbb{C}} W(\mathbb{C}, \sigma) \\
&= W^{(State)}.
\end{aligned}$$

Lemma 163. *For all* $W \in \mathbb{D}_{Prog^n \times State}$ *and* i, $\pi_i(W)^{(Stmt)}(\textbf{\textit{skip}}) \geq W^{(Prog)}(\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}})$.

Proof. For all $W \in \mathbb{D}_{Prog^n \times State}$ and i,

$$\begin{array}{ll} & \pi_i(W)^{(Stmt)}(\textbf{skip}) \\ = & \sum_\sigma \pi_i(W)(\textbf{skip}, \sigma) \\ = & \sum_\sigma \sum_{C_1,\ldots,C_{i-1},C_{i+1},\ldots,C_n} W(C_1 \parallel \cdots \parallel C_{t-1} \parallel \textbf{skip} \parallel C_{t+1} \parallel \cdots \parallel C_n, \sigma) \qquad \text{(by Lem. 160)} \\ \geq & \sum_\sigma W(\textbf{skip} \parallel \cdots \parallel \textbf{skip}, \sigma) \\ = & W^{(Prog)}(\textbf{skip} \parallel \cdots \parallel \textbf{skip}). \end{array}$$

Lemma 164. *For all* W, *if* $W^{(Prog)}(\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}) > 0$, *then* $W|_{\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}} = \lambda(\mathbb{C}, \sigma).\frac{\chi(\mathbb{C}=\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}) \cdot W(\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}, \sigma)}{W^{(Prog)}(\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}})}$.

Proof. For all W such that $W^{(Prog)}(\textbf{skip} \parallel \cdots \parallel \textbf{skip}) > 0$,

$$\begin{array}{rl} W|_{\textbf{skip} \parallel \cdots \parallel \textbf{skip}} & = \lambda(\mathbb{C}, \sigma).\ W|_{\lambda(\mathbb{C},\sigma).\mathbb{C}=\textbf{skip} \parallel \cdots \parallel \textbf{skip}}(\mathbb{C}, \sigma) \\ & = \lambda(\mathbb{C}, \sigma).\ \begin{cases} \frac{W(\mathbb{C},\sigma)}{\mathbf{Pr}_{(\mathbb{C},\sigma)\sim W}[\mathbb{C}=\textbf{skip} \parallel \cdots \parallel \textbf{skip}]}, & \text{if } \mathbb{C} = \textbf{skip} \parallel \cdots \parallel \textbf{skip} \\ 0, & \text{otherwise} \end{cases} \\ & = \lambda(\mathbb{C}, \sigma).\ \frac{\chi(\mathbb{C}=\textbf{skip} \parallel \cdots \parallel \textbf{skip}) \cdot \eta(\textbf{skip} \parallel \cdots \parallel \textbf{skip}, \sigma)}{\mathbf{Pr}_{(\mathbb{C},\sigma)\sim W}[\mathbb{C}=\textbf{skip} \parallel \cdots \parallel \textbf{skip}]} \\ & = \lambda(\mathbb{C}, \sigma).\ \frac{\chi(\mathbb{C}=\textbf{skip} \parallel \cdots \parallel \textbf{skip}) \cdot \eta(\textbf{skip} \parallel \cdots \parallel \textbf{skip}, \sigma)}{W^{(Prog)}(\textbf{skip} \parallel \cdots \parallel \textbf{skip})}. \end{array}$$

Lemma 165. *For all* η, *if* $\eta^{(Stmt)}(\textbf{\textit{skip}}) > 0$, *then* $\eta|_{\textbf{\textit{skip}}} = \lambda(C, \sigma).\frac{\chi(C=\textbf{\textit{skip}}) \cdot \eta(\textbf{\textit{skip}}, \sigma)}{\eta^{(Stmt)}(\textbf{\textit{skip}})}$.

Proof. For all η such that $\eta^{(Stmt)}(\textbf{skip}) > 0$,

$$\begin{array}{rl} \eta|_{\textbf{skip}} & = \lambda(C, \sigma).\ \eta|_{\lambda(C,\sigma).C=\textbf{skip}}(C, \sigma) \\ & = \lambda(C, \sigma).\ \begin{cases} \frac{\eta(C,\sigma)}{\mathbf{Pr}_{(C,\sigma)\sim\eta}[C=\textbf{skip}]}, & \text{if } C = \textbf{skip} \\ 0, & \text{otherwise} \end{cases} \\ & = \lambda(C, \sigma).\ \frac{\chi(C=\textbf{skip}) \cdot \eta(\textbf{skip}, \sigma)}{\mathbf{Pr}_{(\mathbb{C},\sigma)\sim\eta}[C=\textbf{skip}]} \\ & = \lambda(C, \sigma).\ \frac{\chi(C=\textbf{skip}) \cdot \eta(\textbf{skip}, \sigma)}{\eta^{(Stmt)}(\textbf{skip})}. \end{array}$$

Lemma 166. *For all* $W \in \mathbb{D}_{Prog^n \times State}$ *and* i, *if* $W^{(Prog)}(\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}) > 0$,
then $supp(W|_{\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}}{}^{(State)}) \subseteq supp(\pi_i(W)|_{\textbf{\textit{skip}}}{}^{(State)})$.

Proof. For all $W \in \mathbb{D}_{Prog^n \times State}$ and i such that $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$, by Lem. 163 we know $\pi_i(W)^{(Stmt)}(\mathbf{skip}) \geq W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$, thus

$$
\begin{aligned}
& supp(W|_{\mathbf{skip}\parallel\cdots\parallel\mathbf{skip}}{}^{(State)}) \\
&= \{\sigma \mid W|_{\mathbf{skip}\parallel\cdots\parallel\mathbf{skip}}{}^{(State)}(\sigma) > 0\} \\
&= \{\sigma \mid \textstyle\sum_{\mathbb{C}} W|_{\mathbf{skip}\parallel\cdots\parallel\mathbf{skip}}(\mathbb{C},\sigma) > 0\} \\
&= \{\sigma \mid \textstyle\sum_{\mathbb{C}} \frac{\chi(\mathbb{C}=\mathbf{skip}\parallel\cdots\parallel\mathbf{skip})\cdot W(\mathbf{skip}\parallel\cdots\parallel\mathbf{skip},\sigma)}{W^{(Prog)}(\mathbf{skip}\parallel\cdots\parallel\mathbf{skip})} > 0\} \qquad \text{(by Lem. 164)} \\
&= \{\sigma \mid W(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}, \sigma) > 0\} \\
&\subseteq \{\sigma \mid \textstyle\sum_{C_1,\ldots,C_{i-1},C_{i+1},\ldots,C_n} W(C_1 \parallel \cdots \parallel C_{i-1} \parallel \mathbf{skip} \parallel C_{i+1} \parallel \cdots \parallel C_n, \sigma) > 0\} \\
&= \{\sigma \mid \pi_i(W)(\mathbf{skip}, \sigma) > 0\} \qquad \text{(by Lem. 160)} \\
&= \{\sigma \mid \textstyle\sum_{C} \frac{\chi(C=\mathbf{skip})\cdot\pi_i(W)(\mathbf{skip},\sigma)}{\pi_i(W)^{(Stmt)}(\mathbf{skip})} > 0\} \\
&= \{\sigma \mid \textstyle\sum_{C} \pi_i(W)|_{\mathbf{skip}}(C,\sigma) > 0\} \qquad \text{(by Lem. 165)} \\
&= \{\sigma \mid \pi_i(W)|_{\mathbf{skip}}{}^{(State)}(\sigma) > 0\} \\
&= supp(\pi_i(W)|_{\mathbf{skip}}{}^{(State)}).
\end{aligned}
$$

Definition 76. $\Psi \in \mathcal{P}((Prog \times State) \times (Prog \times State))$.

Definition 77. $\Psi^{(State)} \stackrel{\text{def}}{=} \{(\sigma,\sigma') \mid \exists \mathbb{C},\mathbb{C}'.\ ((\mathbb{C},\sigma),(\mathbb{C}',\sigma')) \in \Psi\}$.

Definition 78. Let $\Psi \in \mathcal{P}((Prog^n \times State) \times (Prog^n \times State))$, we define $\pi_i(\Psi) \stackrel{\text{def}}{=} \{((C_i,\sigma),(C_i',\sigma')) \mid \exists C_1,\ldots,C_{t-1},C_{t+1},\ldots,C_n,C_1',\ldots,C_{t-1}',C_{t+1}',\ldots,C_n'.\ ((C_1 \parallel \cdots \parallel C_n),(C_1' \parallel \cdots \parallel C_n')) \in \Psi\}$.

Lemma 167. *For all* $W,W' \in \mathbb{D}_{Prog^n \times State}$, $\Psi \in \mathcal{P}((Prog^n \times State) \times (Prog^n \times State))$ *and* t, *if* $W \overset{t}{\rightsquigarrow} W'$ *and* $\Psi = \{((\mathbb{C},\sigma),(\mathbb{C}',\sigma')) \mid W(\mathbb{C},\sigma) > 0 \wedge (\mathbb{C},\sigma) \overset{p}{\underset{t}{\rightarrow}} (\mathbb{C}',\sigma') \wedge p > 0\}$, *then* $\pi_t(W) \rightsquigarrow (\Psi^{(State)}, \pi_t(W'))$.

Proof. For all $W,W' \in \mathbb{D}_{Prog^n \times State}$, $\Psi \in \mathcal{P}((Prog^n \times State) \times (Prog^n \times State))$ and t such that $W \overset{t}{\rightsquigarrow} W'$ and $\Psi = \{((\mathbb{C},\sigma),(\mathbb{C}',\sigma')) \mid W(\mathbb{C},\sigma) > 0 \wedge (\mathbb{C},\sigma) \overset{p}{\underset{t}{\rightarrow}} (\mathbb{C}',\sigma') \wedge p > 0\}$, we have

$$
\begin{aligned}
& \pi_t(W'') \\
&= \lambda(C',\sigma').\ \mathbf{Pr}_{(C_1'\parallel\cdots\parallel C_n',\sigma'')\sim W''}[C_t' = C'] \\
&= \lambda(C,\sigma').\ \textstyle\sum_{C_1',\ldots,C_n',\sigma''} \{W''(C_1' \parallel \cdots \parallel C_n',\sigma'') \mid C_t' = C'\} \\
&= \lambda(C',\sigma').\ \textstyle\sum_{C_1',\ldots,C_n',\sigma''} \{\sum_{\mathbb{C},\sigma} \{W(\mathbb{C},\sigma)\cdot p \mid (\mathbb{C},\sigma) \overset{p}{\underset{t}{\rightarrow}} (C_1' \parallel \cdots \parallel C_n',\sigma')\} \mid C_t' = C'\} \\
&= \lambda(C',\sigma').\ \textstyle\sum_{C_1',\ldots,C_n',\sigma'',C_1,\ldots,C_n,\sigma} \{W(C_1 \parallel \cdots \parallel C_n,\sigma)\cdot p \mid C_1 = C_1' \wedge \cdots \wedge C_{t-1} = C_{t-1}' \wedge \\
&\qquad C_{t+1} = C_{t+1}' \wedge \cdots \wedge C_n = C_n' \wedge (C_t,\sigma) \overset{p}{\rightarrow} (C_t',\sigma') \wedge C_t' = C'\} \\
&= \lambda(C',\sigma').\ \textstyle\sum_{C_1,\ldots,C_n,\sigma} \{W(C_1 \parallel \cdots \parallel C_n,\sigma)\cdot p \mid (C_t,\sigma) \overset{p}{\rightarrow} (C',\sigma')\} \\
&= \lambda(C',\sigma').\ \textstyle\sum_{C,\sigma'',C_1,\ldots,C_n,\sigma} \{W(C_1 \parallel \cdots \parallel C_n,\sigma)\cdot p \mid (C,\sigma'') \overset{p}{\rightarrow} (C',\sigma') \wedge C_t = C \wedge \sigma = \sigma''\} \\
&= \lambda(C',\sigma').\ \textstyle\sum_{C,\sigma''} \{\sum_{C_1,\ldots,C_n,\sigma} \{W(C_1 \parallel \cdots \parallel C_n,\sigma) \mid C_t = C \wedge \sigma = \sigma''\}\cdot p \mid (C,\sigma'') \overset{p}{\rightarrow} (C',\sigma')\} \\
&= \lambda(C',\sigma').\ \textstyle\sum_{C,\sigma''} \{\mathbf{Pr}_{(C_1\parallel\cdots\parallel C_n,\sigma)\sim W}[C_t = C \wedge \sigma = \sigma'']\cdot p \mid (C,\sigma'') \overset{p}{\rightarrow} (C',\sigma')\} \\
&= \lambda(C',\sigma').\ \textstyle\sum_{C,\sigma''} \{\pi_t(W)\cdot p \mid (C,\sigma'') \overset{p}{\rightarrow} (C',\sigma')\}
\end{aligned}
$$

and

$$
\begin{aligned}
&\Psi^{(State)}\\
&= \{(\sigma,\sigma') \mid \exists \mathbb{C},\mathbb{C}'.\ ((\mathbb{C},\sigma),(\mathbb{C}',\sigma')) \in \Psi\}\\
&= \{(\sigma,\sigma') \mid \exists \mathbb{C},\mathbb{C}'.\ W(\mathbb{C},\sigma) > 0 \wedge (\mathbb{C},\sigma) \xrightarrow[t]{p} (\mathbb{C}',\sigma') \wedge p > 0\}\\
&= \{(\sigma,\sigma') \mid \exists C_1,\ldots,C_n,C'.\ W(C_1 \parallel \cdots \parallel C_n,\sigma) > 0 \wedge (C_t,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
&= \{(\sigma,\sigma') \mid \exists C_1,\ldots,C_n,C',C,\sigma''.\ C_t = C \wedge \sigma = \sigma'' \wedge W(C_1 \parallel \cdots \parallel C_n,\sigma) > 0 \wedge\\
&\qquad (C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
&= \{(\sigma,\sigma') \mid \exists C,C'.\ (\exists C_1,\ldots,C_n,\sigma''.\ C_t = C \wedge \sigma = \sigma'' \wedge W(C_1 \parallel \cdots \parallel C_n,\sigma) > 0) \wedge\\
&\qquad (C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
&= \{(\sigma,\sigma') \mid \exists C,C'.\ \textstyle\sum_{C_1,\ldots,C_n,\sigma''} \{W(C_1 \parallel \cdots \parallel C_n,\sigma) \mid C_t = C \wedge \sigma'' = \sigma\} > 0 \wedge\\
&\qquad (C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
&= \{(\sigma,\sigma') \mid \exists C,C'.\ \mathbf{Pr}_{(C_1\parallel\cdots\parallel C_n,\sigma'')\sim W}[C_t = C \wedge \sigma'' = \sigma] > 0 \wedge (C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
&= \{(\sigma,\sigma') \mid \exists C,C'.\ \pi_t(W)(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\},
\end{aligned}
$$

thus $\pi_t(W) \overset{t}{\rightsquigarrow} (\Psi^{(State)}, \pi_t(W''))$.

Lemma 168. *For all* $W, W' \in \mathbb{D}_{Prog^n \times State}$, $\Psi \in \mathcal{P}((Prog^n \times State) \times (Prog^n \times State))$ *and* t, *if* $W \overset{t}{\rightsquigarrow} W'$ *and* $\Psi = \{((\mathbb{C},\sigma),(\mathbb{C}',\sigma')) \mid W(\mathbb{C},\sigma) > 0 \wedge (\mathbb{C},\sigma) \xrightarrow[t]{p} (\mathbb{C}',\sigma') \wedge p > 0\}$, *then for all* $i \neq t$, $dom(\pi_i(\Psi)) = supp(\pi_i(W))$, $range(\pi_i(\Psi)) = supp(\pi_i(W'))$ *and* $\forall((C_i,\sigma),(C_i',\sigma')) \in \pi_i(\Psi).\ C_i' = C_i \wedge (\sigma,\sigma') \in \Psi^{(State)}$.

Proof. For all $W, W' \in \mathbb{D}_{Prog^n \times State}$, $\Psi \in \mathcal{P}((Prog^n \times State) \times (Prog^n \times State))$ and t such that $W \overset{t}{\rightsquigarrow} W'$ and $\Psi = \{((\mathbb{C},\sigma),(\mathbb{C}',\sigma')) \mid W(\mathbb{C},\sigma) > 0 \wedge (\mathbb{C},\sigma) \xrightarrow[t]{p} (\mathbb{C}',\sigma') \wedge p > 0\}$, for all $i \neq t$, we have

$$
\begin{aligned}
&dom(\pi_i(\Psi))\\
&= \{(C_i,\sigma) \mid \exists C_i',\sigma'.\ ((C_i,\sigma),(C_i',\sigma')) \in \pi_i(\Psi)\}\\
&= \{(C_i,\sigma) \mid \exists C_1,\ldots,C_{i-1},C_{i+1},\ldots,C_n,C_1',\ldots,C_n',\sigma'.\\
&\qquad ((C_1 \parallel \cdots \parallel C_n,\sigma),(C_1' \parallel \cdots \parallel C_n',\sigma')) \in \Psi\}\\
&= \{(C_i,\sigma) \mid \exists C_1,\ldots,C_{i-1},C_{i+1},\ldots,C_n,C_1',\ldots,C_n',\sigma'.\ W(C_1 \parallel \cdots \parallel C_n,\sigma) > 0 \wedge\\
&\qquad (C_1 \parallel \cdots \parallel C_n,\sigma) \xrightarrow[t]{p} (C_1' \parallel \cdots \parallel C_n',\sigma') \wedge p > 0\}\\
&= \{(C_i,\sigma) \mid \exists C_1,\ldots,C_{i-1},C_{i+1},\ldots,C_n.\ W(C_1 \parallel \cdots \parallel C_n,\sigma) > 0\}\\
&= \{(C_i,\sigma) \mid \textstyle\sum_{C_1,\ldots,C_{i-1},C_{i+1},\ldots,C_n} W(C_1 \parallel \cdots \parallel C_n,\sigma) > 0\}\\
&= \{(C_i,\sigma) \mid \pi_i(W)(C_i,\sigma) > 0\} \qquad \text{(by Lem. 160)}\\
&= supp(\pi_i(W)).
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{range}(\pi_i(\Psi))\\
&= \{(C_i', \sigma') \mid \exists C_i, \sigma.\ ((C_i, \sigma), (C_i', \sigma')) \in \pi_i(\Psi)\}\\
&= \{(C_i', \sigma') \mid \exists C_1, \ldots, C_n, C_1', \ldots, C_{i-1}', C_{i+1}', \ldots, C_n', \sigma.\\
&\qquad ((C_1 \parallel \cdots \parallel C_n, \sigma), (C_1' \parallel \cdots \parallel C_n', \sigma')) \in \Psi\}\\
&= \{(C_i', \sigma') \mid \exists C_1, \ldots, C_n, C_1', \ldots, C_{i-1}', C_{i+1}', \ldots, C_n', \sigma.\ W(C_1 \parallel \cdots \parallel C_n, \sigma) > 0 \wedge\\
&\qquad (C_1 \parallel \cdots \parallel C_n, \sigma) \xrightarrow[t]{p} (C_1' \parallel \cdots \parallel C_n', \sigma') \wedge p > 0\}\\
&= \{(C_i', \sigma') \mid \textstyle\sum_{C_1', \ldots, C_{i-1}', C_{i+1}', \ldots, C_n'} \sum_{C_1, \ldots, C_n, \sigma} \{W(C_1 \parallel \cdots \parallel C_n, \sigma) \cdot p \mid\\
&\qquad (C_1 \parallel \cdots \parallel C_n, \sigma) \xrightarrow[t]{p} (C_1' \parallel \cdots \parallel C_n', \sigma')\} > 0\}\\
&= \{(C_i', \sigma') \mid \textstyle\sum_{C_1', \ldots, C_{i-1}', C_{i+1}', \ldots, C_n'} W''(C_1' \parallel \cdots \parallel C_n', \sigma') > 0\} \qquad (\text{from } W \overset{t}{\rightsquigarrow} W'')\\
&= \{(C_i', \sigma') \mid \pi_i(W')(C_i', \sigma') > 0\} \qquad (\text{by Lem. 160})\\
&= \mathit{supp}(\pi_i(W')).
\end{aligned}
$$

For all $C_i, \sigma, C_i', \sigma'$ such that $((C_i, \sigma), (C_i', \sigma')) \in \pi_i(\Psi)$, there exists $C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, C_1', \ldots, C_{t-1}', C_{t+1}', \ldots, C_n'$ such that $((C_1 \parallel \cdots \parallel C_n, \sigma), (C_1' \parallel \cdots \parallel C_n', \sigma')) \in \Psi$, thus $(\sigma, \sigma') \in \Psi^{(State)}$ and $(C_1 \parallel \cdots \parallel C_n, \sigma) \xrightarrow[t]{p} (C_1' \parallel \cdots \parallel C_n', \sigma')$. From $i \neq t$ we know $C_i' = C_i$. Therefore, $\forall((C_i, \sigma), (C_i', \sigma')) \in \pi_i(\Psi).\ C_i' = C_i \wedge (\sigma, \sigma') \in \Psi^{(State)}$.

Lemma 169. *For all $W, W' \in \mathbb{D}_{Prog^n \times State}$ and b, i, if $W|_b = W'$, then $\pi_i(W)|_b = \pi_i(W')$.*

Proof. For all $W, W' \in \mathbb{D}_{Prog^n \times State}$ and b, i such that $W|_b = W'$, by Lem. 82 we know $W' = \lambda(\mathbb{C}, \sigma).\frac{\chi(\sigma \models b) \cdot W(\mathbb{C}, \sigma)}{[\![\mathbf{Pr}(b)]\!]_{W^{(State)}}}$, by Lem. 63 we know $[\![\mathbf{Pr}(b)]\!]_{W^{(State)}} > 0$. By Lem. 162 we know $\pi_i(W)^{(State)} = W^{(State)}$, thus $[\![\mathbf{Pr}(b)]\!]_{\pi_i(W)^{(State)}} = [\![\mathbf{Pr}(b)]\!]_{W^{(State)}} > 0$, so

$$
\begin{aligned}
&\pi_i(W)|_b\\
&= \lambda(C_i, \sigma).\frac{\chi(\sigma \models b) \cdot \pi_i(W)(C_i, \sigma)}{[\![\mathbf{Pr}(b)]\!]_{\pi_i(W)^{(State)}}}\\
&= \lambda(C, \sigma).\frac{\chi(\sigma \models b) \cdot \sum_{C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n} W(C_1 \parallel \cdots \parallel C_n, \sigma)}{[\![\mathbf{Pr}(b)]\!]_{\pi_i(W)^{(State)}}} \qquad (\text{by Lem. 160})\\
&= \lambda(C, \sigma).\textstyle\sum_{C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n} \frac{\chi(\sigma \models b) \cdot W(C_1 \parallel \cdots \parallel C_n, \sigma)}{[\![\mathbf{Pr}(b)]\!]_{W^{(State)}}}\\
&= \lambda(C, \sigma).\textstyle\sum_{C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n} W'(C_1 \parallel \cdots \parallel C_n, \sigma) \qquad (\text{by Lem. 160})\\
&= \pi_i(W').
\end{aligned}
$$

Lemma 170. *For all $W \in \mathbb{D}_{Prog^n \times State}$ and t, $\mathit{nextsplit}(W, t) = \mathit{nextsplit}(\pi_t(W))$.*

Proof. For all $W \in \mathbb{D}_{Prog^n \times State}$ and t,

$$
\begin{aligned}
&\mathit{nextsplit}(W, t)\\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma.\ (C_1 \parallel \cdots \parallel C_n, \sigma) \in \mathit{supp}(W)\}\\
&= \{\mathit{nextsplit}(C_t) \mid \exists C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma.\ W(C_1 \parallel \cdots \parallel C_n, \sigma) > 0\}\\
&= \{\mathit{nextsplit}(C_t) \mid \exists \sigma, C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n.\ W(C_1 \parallel \cdots \parallel C_n, \sigma) > 0\}\\
&= \{\mathit{nextsplit}(C_t) \mid \exists \sigma.\textstyle\sum_{C_1, \ldots, C_{t-1}, C_{t+1}, \ldots, C_n, \sigma} W(C_1 \parallel \cdots \parallel C_n, \sigma) > 0\}\\
&= \{\mathit{nextsplit}(C_t) \mid \exists \sigma.\ \pi_t(W)(C_t, \sigma) > 0\} \qquad (\text{by Lem. 160})\\
&= \{\mathit{nextsplit}(C_t) \mid \exists \sigma.\ (C_t, \sigma) \in \mathit{supp}(\pi_t(W))\}\\
&= \mathit{nextsplit}(\pi_t(W)).
\end{aligned}
$$

Lemma 171. *For all η, $\eta|_{true} = \eta$.*

Proof. For all η, $\eta|_{\text{true}} = \eta|_{\lambda(C,\sigma).\sigma\models\text{true}} = \eta|_{\lambda(C,\sigma).\text{true}} = W$. The last step is by Lem. 5.

Lemma 172. *For all $R_1, \ldots, R_n, G_1, \ldots, G_n$, if $G_j \Rightarrow R_i$ for all i and j such that $i \neq j$, then for all k, φ and $W \in \mathbb{D}_{Prog^n \times State}$, if $(\pi_i(W), R_i, I) \Longmapsto^k_{NST} (G_i, Q_i)$ for all i, then $W \Longmapsto^k_\varphi (I, Q_1 \wedge \cdots \wedge Q_n)$.*

Proof. For all $R_1, \ldots, R_n, G_1, \ldots, G_n$ such that $G_j \Rightarrow R_i$ for all i and j such that $i \neq j$, we prove by induction on k.

- base case: $k = 0$. trivial.
- inductive case: $k = k' + 1$.
  IH: for all φ and $W \in \mathbb{D}_{Prog^n \times State}$, if $(\pi_i(W), R_i, I) \Longmapsto^{k'}_{\text{NST}} (G_i, Q_i)$ for all i, then $W \Longmapsto^{k'}_\varphi (I, Q_1 \wedge \cdots \wedge Q_n)$.
  For all φ and $W \in \mathbb{D}_{Prog^n \times State}$ such that $(\pi_i(W), R_i, I) \Longmapsto^{k'+1}_{\text{NST}} (G_i, Q_i)$ for all i, by definition of *Schedule* there exists t and φ' such that $\varphi = t :: \varphi'$.
  To prove $W \Longmapsto^{k'+1}_\varphi (I, Q_1 \wedge \cdots \wedge Q_n)$, i.e., $W \Longmapsto^{k'+1}_{t::\varphi'} (I, Q_1 \wedge \cdots \wedge Q_n)$, we need to prove
  - $W^{(State)} \models I$.
    From $(\pi_1(W), R_1, I) \Longmapsto^{k'+1}_{\text{NST}} (G_1, Q_1)$ we know $\pi_1(W)^{(State)} \models I$. By Lem. 162 we know $\pi_1(W)^{(State)} = W^{(State)}$, thus $W^{(State)} \models I$.
  - if $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$, then $W|_{\mathbf{skip}\parallel\cdots\parallel\mathbf{skip}}{}^{(State)} \models Q_1 \wedge \cdots \wedge Q_n$.
    For all i, by Lem. 163 we know $\pi_i(W)^{(Stmt)}(\mathbf{skip}) \geq W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$. From $(\pi_i(W), R_i, I) \Longmapsto^{k'+1}_{\text{NST}} (G_i, Q_i)$ we know $\pi_i(W)|_{\mathbf{skip}}{}^{(State)} \models Q_i$. By Lem. 166 we know $supp(W|_{\mathbf{skip}\parallel\cdots\parallel\mathbf{skip}}{}^{(State)}) \subseteq supp(\pi_i(W)|_{\mathbf{skip}}{}^{(State)})$. From $\mathbf{scl}(Q_i)$ we know
    $W|_{\mathbf{skip}\parallel\cdots\parallel\mathbf{skip}}{}^{(State)} \models Q_i$. Therefore $W|_{\mathbf{skip}\parallel\cdots\parallel\mathbf{skip}}{}^{(State)} \models Q_1 \wedge \cdots \wedge Q_n$.
  - for all W', if $W \overset{t}{\hookrightarrow} W'$, then $W' \Longmapsto^{k'}_\varphi (I, Q_1 \wedge \cdots \wedge Q_n)$.

    For all W' such that $W \overset{t}{\hookrightarrow} W'$, there are two cases.

    * there exists $W'', b_1, \ldots, b_k, i$ such that $W \overset{t}{\rightsquigarrow} W''$, $nextsplit(W, t) = \mathbf{split}(b_1, \ldots, b_k)$ and $W''|_{b_i} = W'$.
      Let $\Psi \overset{\text{def}}{=} \{((\mathbb{C}, \sigma), (\mathbb{C}', \sigma')) \mid W(\mathbb{C}, \sigma) > 0 \wedge (\mathbb{C}, \sigma) \xrightarrow[t]{p} (\mathbb{C}', \sigma') \wedge p > 0\}$.
      From $W \overset{t}{\rightsquigarrow} W''$ by Lem. 167 we know $\pi_t(W) \overset{t}{\rightsquigarrow} (\Psi^{(State)}, \pi_t(W''))$.
      By Lem. 170 we know $nextsplit(\pi_t(W))$
      $= nextsplit(W, t) = \mathbf{split}(b_1, \ldots, b_k)$. From $W''|_{b_i} = W'$ by Lem. 169 we know $\pi_t(W'')|_{b_i}$
      $= \pi_t(W')$. From $\pi_t(W) \overset{t}{\rightsquigarrow} (\Psi^{(State)}, \pi_t(W''))$ and $nextsplit(\pi_t(W)) = \mathbf{split}(b_1, \ldots, b_k)$ we know $\pi_t(W) \overset{t}{\hookrightarrow} (\Psi^{(State)}, \pi_t(W'))$. From $(\pi_t(W), R_t, I) \Longmapsto^{k'+1}_{\text{NST}}$

(G_t, Q_t) we know $\Psi^{(State)} \subseteq [\![G_t]\!]$, $\pi_t(W')^{(State)} \models I$ and $(\pi_t(W'), R_t, I) \Longrightarrow^{k'}_{\text{NST}} (G_t, Q_t)$.
For all $i \neq t$, by Lem. 168 we know $dom(\pi_i(\Psi)) = supp(\pi_i(W))$, $range(\pi_i(\Psi)) = supp(\pi_i(W''))$ and $\forall((C_i, \sigma), (C'_i, \sigma')) \in \pi_i(\Psi)$. $C'_i = C_i \wedge (\sigma, \sigma') \in \Psi^{(State)}$. From $i \neq t$ we know $G_t \Rightarrow R_i$, thus $G_t \subseteq [\![R_i]\!]$. From $\Psi^{(State)} \subseteq [\![G_t]\!]$ we know $\Psi^{(State)} \subseteq R_i$, thus $\forall((C_i, \sigma), (C'_i, \sigma')) \in \pi_i(\Psi)$. $C'_i = C_i \wedge (\sigma, \sigma') \models R_i$. From $dom(\pi_i(\Psi)) = supp(\pi_i(W))$ and $range(\pi_i(\Psi)) = supp(\pi_i(W''))$ we know $\pi_i(W) \overset{R_i}{\mapsto} \pi_i(W'')$. From $W''|_{b_i} = W'$ by Lem. 169 we know $\pi_i(W'')|_{b_i} = \pi_i(W')$. By Lem. 162 we know $\pi_i(W')^{(State)} = W'^{(State)} = \pi_t(W')^{(State)}$. From $\pi_t(W')^{(State)} \models I$ we have $\pi_i(W')^{(State)} \models I$. From $\pi_i(W) \overset{R_i}{\mapsto} \pi_i(W'')$ and $\pi_i(W'')|_{b_i} = \pi_i(W')$ we know $\pi_i(W) \overset{R_i}{\underset{I}{\rightarrowtail}} \pi_i(W')$. From $(\pi_i(W), R_i, I) \Longrightarrow^{k'+1}_{\text{NST}} (G_i, Q_i)$ we know $\pi_i(W'), R_i, I \Longrightarrow^{k'}_{\text{NST}} (G_i, Q_i)$.
From $\pi_t(W'), R_t, I \Longrightarrow^{k'}_{\text{NST}} (G_t, Q_t)$ and $\pi_i(W'), R_i, I \Longrightarrow^{k'}_{\text{NST}} (G_i, Q_i)$ for all $i \neq t$ we know $\pi_i(W'), R_i, I \Longrightarrow^{k'}_{\text{NST}} (G_i, Q_i)$ for all i. By IH we have $W' \Longrightarrow^{k'}_{\varphi} (I, Q_1 \wedge \cdots \wedge Q_n)$.

* $W \overset{t}{\rightsquigarrow} W'$ and $|nextsplit(W)| > 1$.

Let $\Psi \overset{\text{def}}{=} \{((\mathbb{C}, \sigma), (\mathbb{C}', \sigma')) \mid W(\mathbb{C}, \sigma) > 0 \wedge (\mathbb{C}, \sigma) \overset{p}{\underset{t}{\rightarrow}} (\mathbb{C}', \sigma') \wedge p > 0\}$.
From $W \overset{t}{\rightsquigarrow} W'$ by Lem. 167 we know $\pi_t(W) \overset{t}{\rightsquigarrow} (\Psi^{(State)}, \pi_t(W'))$.
By Lem. 170 we know $nextsplit(\pi_t(W))$
$= nextsplit(W, t)$, thus $\#nextsplit(\pi_t(W)) = \#nextsplit(W, t) > 1$, therefore $\pi_t(W) \overset{t}{\hookrightarrow} (\Psi^{(State)}, \pi_t(W'))$. From $(\pi_t(W), R_t, I) \Longrightarrow^{k'+1}_{\text{NST}} (G_t, Q_t)$ we know $\Psi^{(State)} \subseteq [\![G_t]\!]$,
$\pi_t(W')^{(State)} \models I$ and $(\pi_t(W'), R_t, I) \Longrightarrow^{k'}_{\text{NST}} (G_t, Q_t)$.
For all $i \neq t$, from $W \overset{t}{\rightsquigarrow} W'$ by Lem. 168 we know $dom(\pi_i(\Psi)) = supp(\pi_i(W))$, $range(\pi_i(\Psi)) = supp(\pi_i(W'))$ and $\forall((C_i, \sigma), (C'_i, \sigma')) \in \pi_i(\Psi)$. $C'_i = C_i \wedge (\sigma, \sigma') \in \Psi^{(State)}$. From $i \neq t$ we know $G_t \Rightarrow R_i$, thus $G_t \subseteq [\![R_i]\!]$. From $\Psi^{(State)} \subseteq [\![G_t]\!]$ we know $\Psi^{(State)} \subseteq R_i$, thus $\forall((C_i, \sigma), (C'_i, \sigma')) \in \pi_i(\Psi)$. $C'_i = C_i \wedge (\sigma, \sigma') \models R_i$. From $dom(\pi_i(\Psi)) = supp(\pi_i(W))$ and $range(\pi_i(\Psi)) = supp(\pi_i(W'))$ we know $\pi_i(W) \overset{R_i}{\mapsto} \pi_i(W')$. By Lem. 171 we know $\pi_i(W')|_{\text{true}} = \pi_i(W')$.
By Lem. 162 we know $\pi_i(W')^{(State)} = W'^{(State)} = \pi_t(W')^{(State)}$.
From $\pi_t(W')^{(State)} \models I$ we have $\pi_i(W')^{(State)} \models I$. From $\pi_i(W) \overset{R_i}{\mapsto} \pi_i(W')$ and $\pi_i(W')|_{\text{true}} = \pi_i(W')$ we know $\pi_i(W) \overset{R_i}{\underset{I}{\rightarrowtail}} \pi_i(W')$. From $(\pi_i(W), R_i, I) \Longrightarrow^{k'+1}_{\text{NST}} (G_i, Q_i)$ we know $\pi_i(W'), R_i, I \Longrightarrow^{k'}_{\text{NST}} (G_i, Q_i)$.
From $\pi_t(W'), R_t, I \Longrightarrow^{k'}_{\text{NST}} (G_t, Q_t)$ and $\pi_i(W'), R_i, I \Longrightarrow^{k'}_{\text{NST}} (G_i, Q_i)$ for all $i \neq t$ we know $\pi_i(W'), R_i, I \Longrightarrow^{k'}_{\text{NST}} (G_i, Q_i)$ for all i. By IH we have $W' \Longrightarrow^{k'}_{\varphi} (I, Q_1 \wedge \cdots \wedge Q_n)$.

Lemma 173. *For all* $C_1,\dots,C_n,P_1,\dots,P_n,Q_1,\dots,Q_n,R_1,\dots,R_n,G_1,\dots,G_n,I$, *if* $R_i,G_i,I \models_{NST} \{P_i\}C_i\{Q_i\}$ *for all* i *and* $G_j \Rightarrow R_i$ *for all* i *and* j *such that* $i \neq j$, *then* $I \models_{ANL} \{P_1 \wedge \cdots \wedge P_n\}C_1 \parallel \cdots \parallel C_n\{Q_1 \wedge \cdots \wedge Q_n\}$.

Proof. For all $C_1,\dots,C_n,P_1,\dots,P_n,Q_1,\dots,Q_n,R_1,\dots,R_n,G_1,\dots,G_n,I$ such that $R_i,G_i,I \models_{\text{NST}} \{P_i\}C_i\{Q_i\}$ for all i and $G_j \Rightarrow R_i$ for all i and j such that $i \neq j$, we need to prove for all μ,φ,k, if $\mu \models I \wedge P_1 \wedge \cdots \wedge P_n$, then $\mathit{init}(C_1 \parallel \cdots \parallel C_n,\mu) \Longmapsto_{\varphi}^{k} (I,Q)$. For all μ,φ,k such that $\mu \models I \wedge P_1 \wedge \cdots \wedge P_n$, from $R_i,G_i,I \models_{\text{NST}} \{P_i\}C_i\{Q_i\}$ for all i we know $(\mathit{init}(C_i,\mu),R_i,I) \Longmapsto_{\text{NST}}^{n} (G_i,Q_i)$ for all i. By Lem. 161 we know $\pi_i(\mathit{init}(C_1 \parallel \cdots \parallel C_n,\mu)) = \mathit{init}(C_i,\mu)$ for all i, thus $(\pi_i(\mathit{init}(C_1 \parallel \cdots \parallel C_n,\mu)),R_i,I) \Longmapsto_{\text{NST}}^{n} (G_i,Q_i)$ for all i. From $G_j \Rightarrow R_i$ for all i and j such that $i \neq j$ by Lem. 172 we know $\mathit{init}(C_1 \parallel \cdots \parallel C_n,\mu) \Longmapsto_{\varphi}^{k} (I,Q_1 \wedge \cdots \wedge Q_n)$.

Lemma 174. *For all* $n,W,\vec{W},\varphi$, *if* $\boldsymbol{History}(W,\varphi,\vec{W})$ *and* $W \Longmapsto_{\varphi}^{n+1} (I,Q)$, *then* $\vec{W}[n]^{(State)} \models I$ *and if* $\vec{W}[n]^{(Prog)}(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}) > 0$, *then* $\vec{W}[n]|_{\boldsymbol{skip}\parallel\cdots\parallel\boldsymbol{skip}}{}^{(State)} \models Q$.

Proof. by induction on n.

- base case: $n = 0$.
  For all $W,\vec{W},\varphi$ such that $\mathbf{History}(W,\varphi,\vec{W})$ and $W \Longmapsto_{\varphi}^{1} (I,Q)$, by Def. 73 we know $W^{(State)} \models I$ and if $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$ then $W|_{\mathbf{skip}\parallel\cdots\parallel\mathbf{skip}}{}^{(State)} \models Q$.
  From $\mathbf{History}(W,\varphi,\vec{W})$ by Lem. 50 we know $\vec{W}[0] = W$, thus $\vec{W}[0]^{(State)} \models I$ and if $\vec{W}[0]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$ then $\vec{W}[0]|_{\mathbf{skip}\parallel\cdots\parallel\mathbf{skip}}{}^{(State)} \models Q$.
- inductive case: $n = k + 1$.
  IH: for all $W,\vec{W},\varphi$, if $\mathbf{History}(W,\varphi,\vec{W})$ and $W \Longmapsto_{\varphi}^{k+1} (I,Q)$, then $\vec{W}[k]^{(State)} \models I$ and if $\vec{W}[k]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$, then $\vec{W}[k]|_{\mathbf{skip}\parallel\cdots\parallel\mathbf{skip}}{}^{(State)} \models Q$.
  For all $W,\vec{W},\varphi$ such that $\mathbf{History}(W,\varphi,\vec{W})$ and $W \Longmapsto_{\varphi}^{n+1} (I,Q)$, from $\mathbf{History}(W,\varphi,\vec{W})$ there exists $t,\varphi',W',\vec{W}'$ such that $\varphi = t :: \varphi'$, $W \overset{t}{\hookrightarrow} W'$, $\mathbf{History}(W',\varphi',\vec{W}')$ and $\vec{W} = W :: \vec{W}'$. From $W \Longmapsto_{\varphi}^{n+1} (I,Q)$ and $\varphi = t :: \varphi'$ we know $W \Longmapsto_{t::\varphi'}^{n+1} (I,Q)$. From $W \overset{t}{\hookrightarrow} W'$ we know $W' \Longmapsto_{\varphi'}^{n} (I,Q)$, i.e., $W' \Longmapsto_{\varphi'}^{k+1} (I,Q)$. From $\mathbf{History}(W',\varphi',\vec{W}')$ by IH we have $\vec{W}'[k]^{(State)} \models I$ and if $\vec{W}'[k]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$, then $\vec{W}'[k]|_{\mathbf{skip}\parallel\cdots\parallel\mathbf{skip}}{}^{(State)} \models Q$.
  From $\vec{W}[n] = (W :: \vec{W}')[k+1] = \vec{W}'[k]$ we know $\vec{W}[n]^{(State)} \models I$ and if $\vec{W}[n]^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) > 0$, then $\vec{W}[n]|_{\mathbf{skip}\parallel\cdots\parallel\mathbf{skip}}{}^{(State)} \models Q$.

Lemma 175. *For all W, if $W^{(Prog)}(\textbf{\textit{skip}} \parallel \cdots \parallel \textbf{\textit{skip}}) = 1$, then $W|_{\textbf{\textit{skip}}\parallel\cdots\parallel\textbf{\textit{skip}}} = W$.*

Proof. For all W such that $W^{(Prog)}(\textbf{skip} \parallel \cdots \parallel \textbf{skip}) = 1$, we have $\mathbf{Pr}_{(\mathbb{C},\sigma)\sim W}[\mathbb{C} = \textbf{skip} \parallel \cdots \parallel \textbf{skip}] = 1$. By Lem. 4 we know $W|_{\lambda(\mathbb{C},\sigma).\mathbb{C}=\textbf{skip}\parallel\cdots\parallel\textbf{skip}} = W$, i.e., $W|_{\textbf{skip}\parallel\cdots\parallel\textbf{skip}} = W$.

Lemma 176. *For all $\mathbb{C}, P, Q, I$, if $I \models_{ANL} \{P\}\mathbb{C}\{Q\}$, $\textbf{lclosed}(I)$ and $\textbf{lclosed}(Q)$, then $\models_{A} \{I \wedge P\}\mathbb{C}\{I \wedge Q\}$.*

Proof. For all $\mathbb{C}, P, Q, I$ such that $I \models_{ANL} \{P\}\mathbb{C}\{Q\}$, $\textbf{lclosed}(I)$ and $\textbf{lclosed}(Q)$, we need to prove for all μ, φ, W such that $\mu \models I \wedge P$ and $\mathit{init}(\mathbb{C}, \mu) \Downarrow_{\varphi} W$, then $W^{(State)} \models I \wedge Q$. For all μ, φ, W such that $\mu \models I \wedge P$ and $\mathit{init}(\mathbb{C}, \mu) \Downarrow_{\varphi} W$, from $I \models_{ANL} \{P\}\mathbb{C}\{Q\}$ and $\mu \models I \wedge P$ we know $\mathit{init}(\mathbb{C}, \mu) \Longrightarrow_{\varphi}^{n} (I, Q)$ for all n. From $\mathit{init}(\mathbb{C}, \mu) \Downarrow_{\varphi} W$ we know there exists $\vec{W}$ such that $\textbf{History}(\mathit{init}(\mathbb{C}, \mu), \varphi, \vec{W})$, $\lim \vec{W} = W$ and $W^{(Prog)}(\textbf{skip} \parallel \cdots \parallel \textbf{skip}) = 1$. From $\textbf{History}(\mathit{init}(\mathbb{C}, \mu), \varphi, \vec{W})$ and $\mathit{init}(\mathbb{C}, \mu) \Longrightarrow_{\varphi}^{n} (I, Q)$ for all n by Lem. 174 we know for all n, $\vec{W}[n]^{(State)} \models I$ and if $\vec{W}[n]^{(Prog)}(\textbf{skip} \parallel \cdots \parallel \textbf{skip}) > 0$, then $\vec{W}[n]|_{\textbf{skip}\parallel\cdots\parallel\textbf{skip}}{}^{(State)} \models Q$. From $\lim \vec{W} = W$ by Lem. 7 we know $\lim \vec{W}^{(Prog)} = W^{(Prog)}$ and $\lim \vec{W}^{(State)} = W^{(State)}$. From $\lim \vec{W}^{(State)} = W^{(State)}$, $\vec{W}^{(State)}[n] = \vec{W}[n]^{(State)} \models I$ for all n, and $\textbf{lclosed}(I)$ we have $W^{(State)} \models I$. From $\lim \vec{W}^{(Prog)} = W^{(Prog)}$ by Lem. 6 we know $\lim_{n\to\infty} \vec{W}[n]^{(Prog)}(\textbf{skip} \parallel \cdots \parallel \textbf{skip}) = 1$. By definition of limit, then there exists N such that $|\vec{W}[n]^{(Prog)}(\textbf{skip} \parallel \cdots \parallel \textbf{skip}) - 1| < 1$ for all $n \geq N$, so $\vec{W}[n]^{(Prog)}(\textbf{skip} \parallel \cdots \parallel \textbf{skip}) > 0$ for all $n \geq N$, thus $\vec{W}[n+N]|_{\textbf{skip}\parallel\cdots\parallel\textbf{skip}}{}^{(State)} \models Q$ for all n. From $\lim \vec{W} = W$ by Lem.10 we know $\lim(\lambda n.\ \vec{W}[n+N]) = W$. From $\vec{W}[n]^{(Prog)}(\textbf{skip} \parallel \cdots \parallel \textbf{skip}) > 0$ for all $n \geq N$ we know $\mathbf{Pr}_{(\mathbb{C},\sigma)\sim\vec{W}[n+N]}(\mathbb{C} = \textbf{skip} \parallel \cdots \parallel \textbf{skip}) > 0$ for all n. From $\textbf{History}(\mathit{init}(\mathbb{C}, \mu), \varphi, \vec{W})$ by Lem. 33 we know $\vec{W}[n+1](\textbf{skip} \parallel \cdots \parallel \textbf{skip}, \sigma) \geq \vec{W}[n](\textbf{skip} \parallel \cdots \parallel \textbf{skip}, \sigma)$ for all n and σ. From $\lim_{n\to\infty} \vec{W}[n]^{(Prog)}(\textbf{skip} \parallel \cdots \parallel \textbf{skip}) = 1$ we know $\lim_{n\to\infty} \mathbf{Pr}_{(\mathbb{C},\sigma)\sim\vec{W}[n+N]}[\mathbb{C} = \textbf{skip} \parallel \cdots \parallel \textbf{skip}] = 1$. From $\lim(\lambda n.\ \vec{W}[n+N]) = W$, $\mathbf{Pr}_{(\mathbb{C},\sigma)\sim\vec{W}[n+N]}(\mathbb{C} = \textbf{skip} \parallel \cdots \parallel \textbf{skip}) > 0$ for all n, $\vec{W}[n+1](\textbf{skip} \parallel \cdots \parallel \textbf{skip}, \sigma) \geq \vec{W}[n](\textbf{skip} \parallel \cdots \parallel \textbf{skip}, \sigma)$ for all n and σ, and $\lim_{n\to\infty} \mathbf{Pr}_{(\mathbb{C},\sigma)\sim\vec{W}[n+N]}[\mathbb{C} = \textbf{skip} \parallel \cdots \parallel \textbf{skip}] = 1$ by Lem. 9 we know $\lim(\lambda n.\ \vec{W}[n+N])|_{\textbf{skip}\parallel\cdots\parallel\textbf{skip}} = W|_{\textbf{skip}\parallel\cdots\parallel\textbf{skip}}$. By Lem. 8 we know $\lim(\lambda n.\ \vec{W}[n+N])|_{\textbf{skip}\parallel\cdots\parallel\textbf{skip}}{}^{(State)} = W|_{\textbf{skip}\parallel\cdots\parallel\textbf{skip}}{}^{(State)}$, i.e.,

$\lim\left(\lambda n.\ \vec{W}[n+N]|_{\mathbf{skip}\|\cdots\|\mathbf{skip}}^{(State)}\right) = W|_{\mathbf{skip}\|\cdots\|\mathbf{skip}}^{(State)}$. From $\mathbf{lclosed}(Q)$ and $\vec{W}[n+N]|_{\mathbf{skip}\|\cdots\|\mathbf{skip}}^{(State)} \models Q$ for all n we know $W|_{\mathbf{skip}\|\cdots\|\mathbf{skip}}^{(State)} \models Q$. From $W^{(Prog)}(\mathbf{skip} \parallel \cdots \parallel \mathbf{skip}) = 1$ by Lem. 175 we know $W^{(State)} = W|_{\mathbf{skip}\|\cdots\|\mathbf{skip}}^{(State)} \models Q$. From $W^{(State)} \models I$ and $W^{(State)} \models Q$ we have $W^{(State)} \models I \wedge Q$.

Lemma 177 (Soundness of (PAR) rule). *For all $C_1, \ldots, C_n, P, Q, I, P_1, \ldots, P_n, Q_1, \ldots, Q_n, R_1, \ldots, R_n, G_1, \ldots, G_n$, if $P \Rightarrow I \wedge P_1 \wedge \cdots \wedge P_n$, $I \wedge Q_1 \wedge \cdots \wedge Q_n \Rightarrow Q$, $\mathbf{lclosed}(I), \mathbf{lclosed}(Q_1), \ldots, \mathbf{lclosed}(Q_n)$, $R_i, G_i, I \models_{\text{NST}} \{P_i\}C_i\{Q_i\}$ for all i, and $G_j \Rightarrow R_i$ for all i and j such that $i \neq j$, then $\models_{\text{A}} \{P\}C_1 \parallel \cdots \parallel C_n\{Q\}$.*

Proof. For all $C_1, \ldots, C_n, P, Q, I, P_1, \ldots, P_n, Q_1, \ldots, Q_n, R_1, \ldots, R_n, G_1, \ldots, G_n$ such that $P \Rightarrow I \wedge P_1 \wedge \cdots \wedge P_n$, $I \wedge Q_1 \wedge \cdots \wedge Q_n \Rightarrow Q$, $\mathbf{lclosed}(I), \mathbf{lclosed}(Q_1), \ldots, \mathbf{lclosed}(Q_n)$, $R_i, G_i, I \models_{\text{NST}} \{P_i\}C_i\{Q_i\}$ for all i, and $G_j \Rightarrow R_i$ for all i and j such that $i \neq j$, from $R_i, G_i, I \models_{\text{NST}} \{P_i\}C_i\{Q_i\}$ for all i, and $G_j \Rightarrow R_i$ for all i and j such that $i \neq j$ by Lem. 173 we know $I \models_{ANL} \{P_1 \wedge \cdots \wedge P_n\}C_1 \parallel \cdots \parallel C_n\{Q_1 \wedge \cdots \wedge Q_n\}$. From $\mathbf{lclosed}(Q_1), \ldots, \mathbf{lclosed}(Q_n)$ we know $\mathbf{lclosed}(Q_1 \wedge \cdots \wedge Q_n)$. By Lem. 176 we know $\models_{\text{A}} \{I \wedge P_1 \wedge \cdots \wedge P_n\}C_1 \parallel \cdots \parallel C_n\{I \wedge Q_1 \wedge \cdots \wedge Q_n\}$. From $P \Rightarrow I \wedge P_1 \wedge \cdots \wedge P_n$ and $I \wedge Q_1 \wedge \cdots \wedge Q_n \Rightarrow Q$ by Lem. 157 we know $\models_{\text{A}} \{P\}C_1 \parallel \cdots \parallel C_n\{Q\}$.

Definition 79. $\psi^{(State)} \stackrel{\text{def}}{=} \{(\sigma, \sigma') \mid \exists C, C'.\ ((C, \sigma), (C', \sigma')) \in \psi\}$.

Lemma 178. *For all R, η, η', if $\eta \overset{R}{\mapsto} \eta'$, then $\eta^{(State)} \overset{R}{\mapsto} \eta'^{(State)}$.*

Proof. For all R, η, η' such that $\eta \overset{R}{\mapsto} \eta'$, there exists ψ such that $dom(\psi) = supp(\eta)$, $range(\psi) = supp(\eta')$ and for all $((C, \sigma), (C', \sigma')) \in \psi$, $C' = C$ and $(\sigma, \sigma') \models R$, thus $\psi^{(State)} = \{(\sigma, \sigma') \mid \exists C, C'.\ ((C, \sigma), (C', \sigma')) \in \psi\} \subseteq \{(\sigma, \sigma') \mid (\sigma, \sigma') \models R\} = [\![R]\!]$,

$$\begin{aligned}
dom(\psi^{(State)}) &= \{\sigma \mid \exists \sigma'.\ (\sigma, \sigma') \in \psi^{(State)}\} \\
&= \{\sigma \mid \exists \sigma', C, C'.\ ((C, \sigma), (C', \sigma')) \in \psi\} \\
&= \{\sigma \mid \exists C.\ (C, \sigma) \in dom(\psi)\} \\
&= \{\sigma \mid \exists C.\ (C, \sigma) \in supp(\eta)\} \\
&= \{\sigma \mid \exists C.\ \eta(C, \sigma) > 0\} \\
&= \{\sigma \mid \textstyle\sum_C \eta(C, \sigma) > 0\} \\
&= \{\sigma \mid \eta^{(State)}(\sigma) > 0\} \\
&= \{\sigma \mid \sigma \in supp(\eta^{(State)})\} \\
&= supp(\eta^{(State)}),
\end{aligned}$$

and

$$\begin{aligned}
range(\psi^{(State)}) &= \{\sigma' \mid \exists\sigma.\ (\sigma,\sigma') \in \psi^{(State)}\} \\
&= \{\sigma' \mid \exists\sigma, C, C'.\ ((C,\sigma),(C',\sigma')) \in \psi\} \\
&= \{\sigma' \mid \exists C'.\ (C',\sigma') \in range(\psi)\} \\
&= \{\sigma' \mid \exists C'.\ (C',\sigma') \in supp(\eta')\} \\
&= \{\sigma' \mid \exists C'.\ \eta'(C',\sigma') > 0\} \\
&= \{\sigma' \mid \textstyle\sum_{C'} \eta'(C',\sigma') > 0\} \\
&= \{\sigma' \mid {\eta'}^{(State)}(\sigma') > 0\} \\
&= \{\sigma' \mid \sigma' \in supp({\eta'}^{(State)})\} \\
&= supp({\eta'}^{(State)}),
\end{aligned}$$

therefore $\eta^{(State)} \overset{R}{\mapsto} {\eta'}^{(State)}$.

Lemma 179. *For all R, I, G, Q, n, η, if $(\eta, R, I) \Longrightarrow^{n}_{\text{ST}} (G, Q)$, then $(\eta, R, I) \Longrightarrow^{n}_{NST} (G, Q)$.*

Proof. For all R, I, G, Q, n, we prove for all η, if $(\eta, R, I) \Longrightarrow^{n}_{\text{ST}} (G, Q)$, then $(\eta, R, I) \Longrightarrow^{n}_{\text{NST}} (G, Q)$ by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all η, if $(\eta, R, I) \Longrightarrow^{k}_{\text{ST}} (G, Q)$, then $(\eta, R, I) \Longrightarrow^{k}_{\text{ST}} (G, Q)$.
  For all η such that $(\eta, R, I) \Longrightarrow^{k+1}_{\text{ST}} (G, Q)$, to prove $(\eta, R, I) \Longrightarrow^{k+1}_{\text{NST}} (G, Q)$, we need to prove
  - if $\eta^{(Stmt)}(\mathbf{skip}) > 0$, then $\eta|_{\mathbf{skip}}{}^{(State)} \models Q$.
    From $(\eta, R, I) \Longrightarrow^{k+1}_{\text{ST}} (G, Q)$ we know if $\eta^{(Stmt)}(\mathbf{skip}) > 0$, then $\eta|_{\mathbf{skip}}{}^{(State)} \models Q$.
  - $\eta^{(State)} \models I$.
    From $(\eta, R, I) \Longrightarrow^{k+1}_{\text{ST}} (G, Q)$ we know $\eta^{(State)} \models I$.
  - for all η', if $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longrightarrow^{k}_{\text{NST}} (G, Q)$.
    For all η' such that $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, from $(\eta, R, I) \Longrightarrow^{k+1}_{\text{ST}} (G, Q)$ we know $(\eta', R, I) \Longrightarrow^{k}_{\text{ST}} (G, Q)$. By IH we have $(\eta', R, I) \Longrightarrow^{k}_{\text{NST}} (G, Q)$.
  - for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, ${\eta'}^{(State)} \models I$ and $(\eta', R, I) \Longrightarrow^{k}_{\text{NST}} (G, Q)$.
    For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, from $(\eta, R, I) \Longrightarrow^{k+1}_{\text{ST}} (G, Q)$ we know $\theta \subseteq [\![G]\!]$, ${\eta'}^{(State)} \models I$ and $(\eta', R, I) \Longrightarrow^{k}_{\text{ST}} (G, Q)$. By IH we have $(\eta', R, I) \Longrightarrow^{k}_{\text{NST}} (G, Q)$.

Lemma 180 (Soundness of (ST-NST) rule). *For all C, R, G, I, P, Q, if $R, G, I \vdash_{ST} \{P\}C\{Q\}$, then $R, G, I \models_{NST} \{P\}C\{Q\}$.*

Proof. For all C, R, G, I, P, Q such that $R, G, I \vdash_{\text{ST}} \{P\}C\{Q\}$, to prove $R, G, I \models_{\text{NST}} \{P\}C\{Q\}$, we need to prove for all μ, if $\mu \models I \wedge P$, then $(init(C, \mu), R, I) \Longrightarrow^{n}_{\text{NST}} (G, Q)$ for all n. For all μ and n such that $\mu \models I \wedge P$, from $R, G, I \vdash_{\text{ST}} \{P\}C\{Q\}$ we know $(init(C, \mu), R, I) \Longrightarrow^{n}_{\text{ST}} (G, Q)$.

Lemma 181. *For all* $R, I, G, Q, R_1, G_1, Q_1, n, \eta$, *if* $R \Rightarrow R_1$, $G_1 \Rightarrow G$, $Q_1 \Rightarrow Q$ *and* $(\eta, R_1, I) \Longmapsto_{\square}^{n} (G_1, Q_1)$, *then* $(\eta, R, I) \Longmapsto_{\square}^{n} (G, Q)$.

Proof. For all $R, I, G, Q, R_1, G_1, Q_1, n$ such that $R \Rightarrow R_1$, $G_1 \Rightarrow G$, $Q_1 \Rightarrow Q$, we prove for all η, if $(\eta, R_1, I) \Longmapsto_{\square}^{n} (G_1, Q_1)$, then $(\eta, R, I) \Longmapsto_{\square}^{n} (G, Q)$ by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all η, if $(\eta, R_1, I) \Longmapsto_{\square}^{k} (G_1, Q_1)$, then $(\eta, R, I) \Longmapsto_{\square}^{k} (G, Q)$.
  For all η such that $(\eta, R_1, I) \Longmapsto_{\square}^{k+1} (G_1, Q_1)$, to prove $(\eta, R, I) \Longmapsto_{\square}^{k+1} (G, Q)$, we need to prove
  - (when $\square = \text{ST}$) $\eta^{(Stmt)}(\mathbf{skip}) = 0$ or $\eta^{(Stmt)}(\mathbf{skip}) = 1$.
    From $(\eta, R_1, I) \Longmapsto_{\text{ST}}^{k+1} (G_1, Q_1)$ we know $\eta^{(Stmt)}(\mathbf{skip}) = 0$ or $\eta^{(Stmt)}(\mathbf{skip}) = 1$.
  - if $\eta^{(Stmt)}(\mathbf{skip}) > 0$, then $\eta|_{\mathbf{skip}}{}^{(State)} \models Q$.
    From $(\eta, R_1, I) \Longmapsto_{\square}^{k+1} (G_1, Q_1)$ and $\eta^{(Stmt)}(\mathbf{skip}) > 0$ we know $\eta|_{\mathbf{skip}}{}^{(State)} \models Q_1$. From $Q_1 \Rightarrow Q$ we know $\eta|_{\mathbf{skip}}{}^{(State)} \models Q$.
  - $\eta^{(State)} \models I$.
    From $(\eta, R_1, I) \Longmapsto_{\square}^{k+1} (G_1, Q_1)$ we know $\eta^{(State)} \models I$.
  - for all η', if $\eta \underset{I}{\overset{R}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longmapsto_{\square}^{k} (G, Q)$.
    For all η' such that $\eta \underset{I}{\overset{R}{\rightarrowtail}} \eta'$, from $(\eta, R_1, I) \Longmapsto_{\square}^{k+1} (G_1, Q_1)$ we know $(\eta', R_1, I) \Longmapsto_{\square}^{k} (G_1, Q_1)$. By IH we have $(\eta', R, I) \Longmapsto_{\square}^{k} (G, Q)$.
  - for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, $\eta'^{(State)} \models I$ and $(\eta', R, I) \Longmapsto_{\square}^{k} (G, Q)$.
    For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, from $(\eta, R_1, I) \Longmapsto_{\square}^{k+1} (G_1, Q_1)$ we know $\theta \subseteq [\![G_1]\!]$, $\eta'^{(State)} \models I$ and $(\eta', R_1, I) \Longmapsto_{\square}^{k} (G_1, Q_1)$. From $\theta \subseteq [\![G_1]\!]$ and $G_1 \Rightarrow G$ we know $\theta \subseteq [\![G_1]\!] \subseteq [\![G]\!]$. From $(\eta', R_1, I) \Longmapsto_{\square}^{k} (G_1, Q_1)$ by IH we know By IH we have $(\eta', R, I) \Longmapsto_{\square}^{k} (G, Q)$.

Lemma 182 (Soundness of (CSQ) rule). *For all* $C, I, R, G, P, Q, R_1, G_1, P_1, Q_1$, *if* $P \Rightarrow P_1$, $R \Rightarrow R_1$, $G_1 \Rightarrow G$, $Q_1 \Rightarrow Q$ *and* $R_1, G_1, I \models_{\square} \{P_1\} C \{Q_1\}$, *then* $R, G, I \models_{\square} \{P\} C \{Q\}$.

Proof. For all $C, I, R, G, P, Q, R_1, G_1, P_1, Q_1$ such that $P \Rightarrow P_1$, $R \Rightarrow R_1$, $G_1 \Rightarrow G$, $Q_1 \Rightarrow Q$ and $R_1, G_1, I \models_{\square} \{P_1\} C \{Q_1\}$, to prove $R, G, I \models_{\square} \{P\} C \{Q\}$, we need to prove for all μ, if $\mu \models P$, then $(\mathit{init}(C, \mu), R, I) \Longmapsto_{\square}^{n} (G, Q)$ for all n. For all μ and n such that $\mu \models P$, from $P \Rightarrow P_1$ we know $\mu \models P_1$. From $R_1, G_1, I \models_{\square} \{P_1\} C \{Q_1\}$ we know $(\mathit{init}(C, \mu), R_1, I) \Longmapsto_{\square}^{n} (G_1, Q_1)$. From $R \Rightarrow R_1$, $G_1 \Rightarrow G$ and $Q_1 \Rightarrow Q$ by Lem. 181 we know $(\mathit{init}(C, \mu), R, I) \Longmapsto_{\square}^{n} (G, Q)$.

Lemma 183 (Soundness of (DISJ) rule). *For all* $C, R, G, I, P_1, P_2, Q_1, Q_2$, *if* $R, G, I \models_{\square} \{P_1\} C \{Q_1\}$ *and* $R, G, I \models_{\square} \{P_2\} C \{Q_2\}$, *then* $R, G, I \models_{\square} \{P_1 \vee P_2\} C \{Q_1 \vee P_2\}$.

Proof. For all $C, R, G, I, P_1, P_2, Q_1, Q_2$ such that $R, G, I \models_{\Box} \{P_1\}C\{Q_1\}$ and $R, G, I \models_{\Box} \{P_2\}C\{Q_2\}$, to prove $R, G, I \models_{\Box} \{P_1 \vee P_2\}C\{Q_1 \vee P_2\}$, we need to prove for all μ, if $\mu \models I \wedge (P_1 \vee P_2)$, then $(\mathit{init}(C, \mu), R, I) \Longrightarrow^n_{\Box} (G, Q_1 \vee Q_2)$. For all μ such that $\mu \models I \wedge (P_1 \vee P_2)$, we know $\mu \models I \wedge P_1$ or $\mu \models I \wedge P_2$. We prove the two cases respectively.

- case 1: $\mu \models I \wedge P_1$.
  From $\mu \models I \wedge P_1$ and $R, G, I \models_{\Box} \{P_1\}C\{Q_1\}$ we know $(\mathit{init}(C, \mu), R, I) \Longrightarrow^n_{\Box} (G, Q_1)$. From $Q_1 \Rightarrow Q_1 \vee Q_2$ by Lem. 181 we know $(\mathit{init}(C, \mu), R, I) \Longrightarrow^n_{\Box} (G, Q_1 \vee Q_2)$.
- case 2: $\mu \models I \wedge P_2$.
  From $\mu \models I \wedge P_2$ and $R, G, I \models_{\Box} \{P_2\}C\{Q_2\}$ we know $(\mathit{init}(C, \mu), R, I) \Longrightarrow^n_{\Box} (G, Q_2)$. From $Q_2 \Rightarrow Q_1 \vee Q_2$ by Lem. 181 we know $(\mathit{init}(C, \mu), R, I) \Longrightarrow^n_{\Box} (G, Q_1 \vee Q_2)$.

Lemma 184. *For all $R, I, G, Q_1, Q_2, n, \eta$, if $(\eta, R, I) \Longrightarrow^n_{\Box} (G, Q_1)$ and $(\eta, R, I) \Longrightarrow^n_{\Box} (G, Q_2)$, then $(\eta, R, I) \Longrightarrow^n_{\Box} (G, Q_1 \wedge Q_2)$.*

Proof. For all R, I, G, Q_1, Q_2, n, we prove for all η, if $(\eta, R, I) \Longrightarrow^n_{\Box} (G, Q_1)$ and $(\eta, R, I) \Longrightarrow^n_{\Box} (G, Q_2)$, then $(\eta, R, I) \Longrightarrow^n_{\Box} (G, Q_1 \wedge Q_2)$ by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all η, if $(\eta, R, I) \Longrightarrow^k_{\Box} (G, Q_1)$ and $(\eta, R, I) \Longrightarrow^n_{\Box} (G, Q_2)$, then $(\eta, R, I) \Longrightarrow^k_{\Box} (G, Q_1 \wedge Q_2)$.
  For all η such that $(\eta, R, I) \Longrightarrow^{k+1}_{\Box} (G, Q_1)$ and $(\eta, R, I) \Longrightarrow^n_{\Box} (G, Q_2)$, to prove $(\eta, R, I) \Longrightarrow^{k+1}_{\Box} (G, Q_1 \wedge Q_2)$, we need to prove
  - (when $\Box = \text{ST}$) $\eta^{(Stmt)}(\mathbf{skip}) = 0$ or $\eta^{(Stmt)}(\mathbf{skip}) = 1$.
    From $(\eta, R, I) \Longrightarrow^{k+1}_{\text{ST}} (G, Q_1)$ we know $\eta^{(Stmt)}(\mathbf{skip}) = 0$ or $\eta^{(Stmt)}(\mathbf{skip}) = 1$.
  - if $\eta^{(Stmt)}(\mathbf{skip}) > 0$, then $\eta|_{\mathbf{skip}}{}^{(State)} \models Q_1 \wedge Q_2$.
    From $(\eta, R, I) \Longrightarrow^{k+1}_{\Box} (G, Q_1)$ and $\eta^{(Stmt)}(\mathbf{skip}) > 0$ we know $\eta|_{\mathbf{skip}}{}^{(State)} \models Q_1$. From $(\eta, R, I) \Longrightarrow^{k+1}_{\Box} (G, Q_2)$ and $\eta^{(Stmt)}(\mathbf{skip}) > 0$ we know $\eta|_{\mathbf{skip}}{}^{(State)} \models Q_2$. Therefore $\eta|_{\mathbf{skip}}{}^{(State)} \models Q_1 \wedge Q_2$.
  - $\eta^{(State)} \models I$.
    From $(\eta, R, I) \Longrightarrow^{k+1}_{\Box} (G, Q_1)$ we know $\eta^{(State)} \models I$.
  - for all η', if $\eta \xrightarrow[I]{R} \eta'$, then $(\eta', R, I) \Longrightarrow^k_{\Box} (G, Q)$.
    For all η' such that $\eta \xrightarrow[I]{R} \eta'$, from $(\eta, R, I) \Longrightarrow^{k+1}_{\Box} (G, Q_1)$ we know $(\eta', R, I) \Longrightarrow^k_{\Box} (G, Q_1)$. From $(\eta, R, I) \Longrightarrow^{k+1}_{\Box} (G, Q_2)$ we know $(\eta', R, I) \Longrightarrow^k_{\Box} (G, Q_2)$. By IH we have
    $(\eta', R, I) \Longrightarrow^k_{\Box} (G, Q_1 \wedge Q_2)$.
  - for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, $\eta'^{(State)} \models I$ and $(\eta', R, I) \Longrightarrow^k_{\Box} (G, Q_1 \wedge Q_2)$.
    For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, from $(\eta, R, I) \Longrightarrow^{k+1}_{\Box} (G, Q_1)$

we know $\theta \subseteq [\![G]\!]$, ${\eta'}^{(State)} \models I$ and $(\eta', R, I) \Longrightarrow_{\square}^{k} (G, Q_1)$. From $(\eta, R, I) \Longrightarrow_{\square}^{k+1} (G, Q_1)$ we know $(\eta', R, I) \Longrightarrow_{\square}^{k} (G, Q_2)$. By IH we have $(\eta', R, I) \Longrightarrow_{\square}^{k} (G, Q_1 \wedge Q_2)$.

Lemma 185 (Soundness of (CONJ) rule). *For all $C, R, G, I, P_1, P_2, Q_1, Q_2$, if $R, G, I \models_{\square} \{P_1\}C\{Q_1\}$ and $R, G, I \models_{\square} \{P_2\}C\{Q_2\}$, then $R, G, I \models_{\square} \{P_1 \wedge P_2\}C\{Q_1 \wedge P_2\}$.*

Proof. For all $C, R, G, I, P_1, P_2, Q_1, Q_2$ such that $R, G, I \models_{\square} \{P_1\}C\{Q_1\}$ and $R, G, I \models_{\square} \{P_2\}C\{Q_2\}$, to prove $R, G, I \models_{\square} \{P_1 \wedge P_2\}C\{Q_1 \wedge P_2\}$, we need to prove for all μ, if $\mu \models I \wedge (P_1 \wedge P_2)$, then $(\mathit{init}(C, \mu), R, I) \Longrightarrow_{\square}^{n} (G, Q_1 \wedge Q_2)$. For all μ such that $\mu \models I \wedge (P_1 \vee P_2)$, we know $\mu \models I \wedge P_1$ and $\mu \models I \wedge P_2$. From $\mu \models I \wedge P_1$ and $R, G, I \models_{\square} \{P_1\}C\{Q_1\}$ we know $(\mathit{init}(C, \mu), R, I) \Longrightarrow_{\square}^{n} (G, Q_1)$. From $\mu \models I \wedge P_2$ and $R, G, I \models_{\square} \{P_2\}C\{Q_2\}$ we know $(\mathit{init}(C, \mu), R, I) \Longrightarrow_{\square}^{n} (G, Q_2)$. By Lem. 184 we know $(\mathit{init}(C, \mu), R, I) \Longrightarrow_{\square}^{n} (G, Q_1 \wedge Q_2)$

Lemma 186. *For all Q, R, I, η, η', if $\mathbf{Sta}(Q, R, I)$, ${\eta}^{(State)} \models I \wedge Q$ and $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then ${\eta'}^{(State)} \models I \wedge Q$.*

Proof. For all Q, R, I, η, η' such that $\mathbf{Sta}(Q, R, I)$, ${\eta}^{(State)} \models I \wedge Q$ and $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, from $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$ there exists η'' and b such that $\eta \overset{R}{\mapsto} \eta''$, $\eta''|_b = \eta'$ and ${\eta'}^{(State)} \models I$. From $\eta \overset{R}{\mapsto} \eta''$ by Lem. 178 we know ${\eta}^{(State)} \overset{R}{\mapsto} {\eta''}^{(State)}$. From $\eta''|_b = \eta'$ by Lem. 20 we have $\mathit{supp}(\eta') \subseteq \mathit{supp}(\eta'')$. By Lem. 24 we know $\mathit{supp}({\eta'}^{(State)}) \subseteq \mathit{supp}({\eta''}^{(State)})$. From ${\eta}^{(State)} \models I$, ${\eta'}^{(State)} \models I$ and ${\eta}^{(State)} \overset{R}{\mapsto} {\eta''}^{(State)}$ we have ${\eta}^{(State)} \overset{R}{\underset{I}{\rightarrowtail}} {\eta'}^{(State)}$. From $\mathbf{Sta}(Q, R, I)$ and ${\eta}^{(State)} \models Q$ we have ${\eta'}^{(State)} \models Q$.

Lemma 187. *For all R, η, η', if $\eta \overset{R}{\mapsto} \eta'$, then $\mathit{supp}({\eta'}^{(Stmt)}) = \mathit{supp}({\eta}^{(Stmt)})$.*

Proof. For all R, η, η' such that $\eta \overset{R}{\mapsto} \eta'$, there exists ψ such that $\mathit{dom}(\psi) = \mathit{supp}(\eta)$, $\mathit{range}(\psi) = \mathit{supp}(\eta')$ and for all $((C, \sigma), (C', \sigma')) \in \psi$, $C' = C$ and $(\sigma, \sigma') \models R$, thus

$$
\begin{aligned}
\mathit{supp}({\eta'}^{(Stmt)}) &= \mathit{dom}(\mathit{supp}(\eta')) && \text{(by Lem. 21)} \\
&= \mathit{dom}(\mathit{range}(\psi)) \\
&= \{C' \mid \exists \sigma'.\ (C', \sigma') \in \mathit{range}(\psi)\} \\
&= \{C' \mid \exists \sigma', C, \sigma.\ ((C, \sigma), (C', \sigma')) \in \psi\} \\
&= \{C' \mid \exists \sigma', C, \sigma.\ ((C, \sigma), (C', \sigma')) \in \psi \wedge C' = C\} \\
&= \{C \mid \exists \sigma', C', \sigma.\ ((C, \sigma), (C', \sigma')) \in \psi\} \\
&= \{C \mid \exists \sigma.\ (C, \sigma) \in \mathit{dom}(\psi)\} \\
&= \mathit{dom}(\mathit{dom}(\psi)) \\
&= \mathit{dom}(\mathit{supp}(\eta)) \\
&= \mathit{supp}({\eta}^{(Stmt)}). && \text{(by Lem. 21)}
\end{aligned}
$$

Lemma 188. *For all R, I, η, η', if $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $supp(\eta'^{(Stmt)}) \subseteq supp(\eta^{(Stmt)})$.*

Proof. For all R, I, η, η' such that $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, there exists η'' and b such that $\eta \overset{R}{\mapsto} \eta''$, $\eta''|_b = \eta'$ and $\eta'^{(State)} \models I$. From $\eta \overset{R}{\mapsto} \eta''$ by Lem. 187 we know $supp(\eta''^{(Stmt)}) = supp(\eta^{(Stmt)})$. From $\eta''|_b = \eta'$ by Lem. 20 we know $supp(\eta') \subseteq supp(\eta'')$. By Lem. 23 we know $supp(\eta'^{(Stmt)}) \subseteq supp(\eta''^{(Stmt)})$, thus $supp(\eta'^{(Stmt)}) \subseteq supp(\eta^{(Stmt)})$.

Lemma 189. *For all η and C_1, $\eta^{(Stmt)} = \delta(C_1)$ if and only if $\forall (C, \sigma) \in supp(\eta).\ C = C_1$.*

Proof. For all η and C_1, we have

$$
\begin{array}{ll}
 & \eta^{(Stmt)} = \delta(C_1) \\
\iff & supp(\eta^{(Stmt)}) = \{C_1\} \qquad \text{(by Lem. 26)} \\
\iff & dom(supp(\eta)) = \{C_1\} \qquad \text{(by Lem. 21)} \\
\iff & \{C \mid \exists \sigma.\ (C, \sigma) \in supp(\eta)\} = \{C_1\} \\
\iff & \{C \mid \exists \sigma.\ (C, \sigma) \in supp(\eta)\} \subseteq \{C_1\} \\
\iff & \forall (C, \sigma) \in supp(\eta).\ C = C_1.
\end{array}
$$

Lemma 190. *For all η and C_1, if $\eta^{(Stmt)} = \delta(C_1)$, then $nextsplit(\eta) = \{nextsplit(C_1)\}$.*

Proof. For all η and C_1 such that $\eta^{(Stmt)} = \delta(C_1)$,

$$
\begin{array}{ll}
 & nextsplit(\eta) \\
= & \{nextsplit(C) \mid \exists \sigma.\ (C, \sigma) \in supp(\eta)\} \\
= & \{nextsplit(C) \mid \exists \sigma.\ (C, \sigma) \in supp(\eta) \wedge C = C_1\} \qquad \text{(by Lem. 189)} \\
= & \{nextsplit(C_1) \mid \exists \sigma.\ (C, \sigma) \in supp(\eta) \wedge C = C_1\} \\
= & \{nextsplit(C_1)\}.
\end{array}
$$

Lemma 191. *For all η and η', if $nextsplit(\eta) \supseteq \{\textbf{\textit{split}}(true)\}$, then $\eta \rightsquigarrow \eta'$ if and only if $\eta \hookrightarrow \eta'$.*

Proof. For all η and η' such that $nextsplit(\eta) \supseteq \{\textbf{split}(\text{true})\}$, we prove the two directions respectively.

- if $\eta \rightsquigarrow \eta'$, from $nextsplit(\eta) \supseteq \{\textbf{split}(\text{true})\}$ we know $nextsplit(\eta) = \{\textbf{split}(\text{true})\}$ or
  $nextsplit(\eta) \supset \{\textbf{split}(\text{true})\}$. We prove the two cases respectively.
  - $nextsplit(\eta) = \{\textbf{split}(\text{true})\}$.
    By Lem. 171 we know $\eta'|_{\text{true}} = \eta'$. From $\eta \rightsquigarrow \eta'$, $nextsplit(\eta) = \{\textbf{split}(\text{true})\}$ and $\eta'|_{\text{true}} = \eta'$ we have $\eta \hookrightarrow \eta'$.
  - $nextsplit(W, t) \supset \{\textbf{split}(\text{true})\}$.
    $\#nextsplit(W, t) > 1$, so $W \overset{t}{\hookrightarrow} W'$.
- if $\eta \hookrightarrow \eta'$, there are two cases.

- case 1: there exists $\eta'', b_1, \ldots, b_k, i$ such that $\eta \rightsquigarrow \eta''$, $\mathit{nextsplit}(\eta) = \{\mathbf{split}(b_1, \ldots, b_k)\}$ and $\eta''|_{b_i} = \eta'$.
  From $\mathit{nextsplit}(\eta) \supseteq \{\mathbf{split}(\mathrm{true})\}$ we know $k = i = 1$, $b_1 = \mathrm{true}$. By Lem. 171 we know $\eta''|_{\mathrm{true}} = \eta''$, so $\eta' = \eta''|_{b_i} = \eta''|_{\mathrm{true}} = \eta''$. From $\eta \rightsquigarrow \eta''$ we have $\eta \rightsquigarrow \eta'$.
- case 2: $\#\mathit{nextsplit}(\eta) > 1$ and $\eta \rightsquigarrow \eta'$. trivial.

Lemma 192. *For all η, if $\eta^{(Stmt)} = \delta(\boldsymbol{skip})$, then $\eta \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}, \delta(\boldsymbol{skip}) \otimes \eta^{(State)})$.*

Proof. For all η such that $\eta^{(Stmt)} = \delta(\mathbf{skip})$, by Lem. 189 we know $C = \mathbf{skip}$ for all $(C, \sigma) \in \mathit{supp}(\eta)$, thus

$$
\begin{aligned}
&\lambda(C', \sigma'). \textstyle\sum_{C,\sigma} \{\eta(C, \sigma) \cdot p \mid (C, \sigma) \xrightarrow{p} (C', \sigma')\} \\
&= \lambda(C', \sigma'). \textstyle\sum_{C,\sigma} \{\eta(C, \sigma) \cdot p \mid (C, \sigma) \in \mathit{supp}(\eta) \wedge (C, \sigma) \xrightarrow{p} (C', \sigma')\} \\
&= \lambda(C', \sigma'). \textstyle\sum_{C,\sigma} \{\eta(C, \sigma) \cdot p \mid (C, \sigma) \in \mathit{supp}(\eta) \wedge C = \mathbf{skip} \wedge (C, \sigma) \xrightarrow{p} (C', \sigma')\} \\
&= \lambda(C', \sigma'). \textstyle\sum_{\sigma} \{\eta^{(State)}(\sigma) \cdot p \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge (\mathbf{skip}, \sigma) \xrightarrow{p} (C', \sigma')\} \\
&= \lambda(C', \sigma'). \textstyle\sum_{\sigma} \{\eta^{(State)}(\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge C' = \mathbf{skip} \wedge \sigma' = \sigma\} \\
&= \lambda(C', \sigma'). \textstyle\sum_{\sigma} \{\eta^{(State)}(\sigma) \mid C' = \mathbf{skip} \wedge \sigma' = \sigma\} \\
&= \lambda(C', \sigma').\ \delta(\mathbf{skip})(C') \cdot \eta^{(State)}(\sigma') \\
&= \delta(\mathbf{skip}) \otimes \eta^{(State)}
\end{aligned}
$$

and

$$
\begin{aligned}
&\{(\sigma, \sigma') \mid \exists C, C'.\ \eta(C, \sigma) > 0 \wedge (C, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\
&= \{(\sigma, \sigma') \mid \exists C, C'.\ (C, \sigma) \in \mathit{supp}(\eta) \wedge (C, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\
&= \{(\sigma, \sigma') \mid \exists C, C'.\ (C, \sigma) \in \mathit{supp}(\eta) \wedge (C = \mathbf{skip}) \wedge (C, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\
&= \{(\sigma, \sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \exists C'.\ (\mathbf{skip}, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\
&= \{(\sigma, \sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \exists C'.\ C' = \mathbf{skip} \wedge \sigma' = \sigma\} \\
&= \{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}.
\end{aligned}
$$

Therefore $\eta \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}, \delta(\mathbf{skip}) \otimes \eta^{(State)})$.

Lemma 193. *For all $\eta, \theta_1, \eta_1, \theta_2, \eta_2$, if $\eta \rightsquigarrow (\theta_1, \eta_1)$ and $\eta \rightsquigarrow (\theta_2, \eta_2)$, then $\theta_1 = \theta_2$ and $\eta_1 = \eta_2$.*

Proof. For all $\eta, \theta_1, \eta_1, \theta_2, \eta_2$ such that $\eta \rightsquigarrow (\theta_1, \eta_1)$ and $\eta \rightsquigarrow (\theta_2, \eta_2)$, from $\eta \rightsquigarrow (\theta_1, \eta_1)$ we know $\eta_1 = \lambda(C', \sigma'). \sum_{C,\sigma} \{\eta(C, \sigma) \cdot p \mid (C, \sigma) \xrightarrow{p} (C', \sigma')\}$ and $\theta_1 = \{(\sigma, \sigma') \mid \exists C, C'.\ (C, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\}$. From $\eta \rightsquigarrow (\theta_2, \eta_2)$ we know $\eta_2 = \lambda(C', \sigma'). \sum_{C,\sigma} \{\eta(C, \sigma) \cdot p \mid (C, \sigma) \xrightarrow{p} (C', \sigma')\}$ and $\theta_2 = \{(\sigma, \sigma') \mid \exists C, C'.\ (C, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\}$. Therfore $\theta_1 = \theta_2$ and $\eta_1 = \eta_2$.

Lemma 194. *For all Q, R, G, I, if $\mathbf{Sta}(Q, R, I)$ and $\mathbf{Id} \Rightarrow G$, then for all n and η, if $\eta^{(Stmt)} = \delta(\boldsymbol{skip})$ and $\eta^{(State)} \models I \wedge Q$, then $(\eta, R, I) \Longmapsto^n_{ST} (G, Q)$.*

Proof. For all Q, R, G, I, if $\mathbf{Sta}(Q, R, I)$ and $\mathbf{Id} \Rightarrow G$, we prove by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all η, if $\eta^{(Stmt)} = \delta(\mathbf{skip})$ and $\eta^{(State)} \models I \wedge Q$, then $(\eta, R, I) \Longmapsto^{k}_{\mathrm{ST}} (G, Q)$.
  For all η such that $\eta^{(Stmt)} = \delta(\mathbf{skip})$ and $\eta^{(State)} \models I \wedge Q$, to prove $(\eta, R, I) \Longmapsto^{k+1}_{\mathrm{ST}} (G, Q)$, by Def. 33 we need to prove
  - $\eta^{(Stmt)}(\mathbf{skip}) = 0$ or $\eta^{(Stmt)}(\mathbf{skip}) = 1$.
    by assumption we know $\eta^{(Stmt)}(\mathbf{skip}) = 1$.
  - if $\eta^{(Stmt)}(\mathbf{skip}) > 0$, then $\eta^{(State)} \models Q$.
    From $\eta^{(Stmt)}(\mathbf{skip}) = 1$ we know $\mathbf{Pr}_{(C,\sigma)\sim\eta}[C = \mathbf{skip}] = 1$. By Lem. 4 we know $\eta|_{\mathbf{skip}} = \eta|_{\lambda(C,\sigma).C=\mathbf{skip}} = \eta$. From $\eta^{(State)} \models I \wedge Q$ we know ${\eta|_{\mathbf{skip}}}^{(State)} \models Q$.
  - $\eta^{(State)} \models I$.
    From $\eta^{(State)} \models I \wedge Q$ we know $\eta^{(State)} \models I$.
  - for all η', if $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longmapsto^{k}_{\mathrm{ST}} (G, Q)$.
    For all η' such that $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, by Lem. 188 we know $supp({\eta'}^{(Stmt)}) \subseteq supp(\eta^{(Stmt)})$. From $\eta^{(Stmt)} = \delta(\mathbf{skip})$ by Lem. 27 we know ${\eta'}^{(Stmt)} = \delta(\mathbf{skip})$. From $\mathbf{Sta}(Q, R, I)$, $\eta^{(State)} \models I \wedge Q$ and $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$ by Lem. 186 we have ${\eta'}^{(State)} \models I \wedge Q$. From ${\eta'}^{(Stmt)} = \delta(\mathbf{skip})$ by IH we have $(\eta', R, I) \Longmapsto^{k}_{\mathrm{ST}} (G, Q)$.
  - for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, ${\eta'}^{(State)} \models I$ and $(\eta', R, I) \Longmapsto^{k}_{\mathrm{ST}} (G, Q)$.
    For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, from $\eta^{(Stmt)} = \delta(\mathbf{skip})$ by Lem. 190 we have $nextsplit(\eta) = \{nextsplit(\mathbf{skip})\} = \{\mathbf{split}(\mathrm{true})\}$. From $\eta \hookrightarrow (\theta, \eta')$ by Lem. 191 we know $\eta \rightsquigarrow (\theta, \eta')$. From $\eta^{(Stmt)} = \delta(\mathbf{skip})$ by Lem. 192 we know $\eta \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in supp(\eta^{(State)})\}, \delta(\mathbf{skip}) \otimes \eta^{(State)})$. From $\eta \rightsquigarrow (\theta, \eta')$ by Lem. 193 we know $\theta = \{(\sigma, \sigma) \mid \sigma \in supp(\eta^{(State)})\}$ and $\eta' = \delta(\mathbf{skip}) \otimes \eta^{(State)}$, thus $\theta \subseteq [\![\mathbf{Id}]\!] \subseteq [\![G]\!]$. From $\eta' = \delta(\mathbf{skip}) \otimes \eta^{(State)}$ by Lem. 18 and Lem. 19 we know ${\eta'}^{(Stmt)} = \delta(\mathbf{skip})$ and ${\eta'}^{(State)} = \eta^{(State)} \models I \wedge Q$. By IH we have $(\eta', R, I) \Longmapsto^{k}_{\mathrm{ST}} (G, Q)$.

Lemma 195 (Soundness of (SKIP) rule). *For all Q, R, G, I, if $\mathbf{Sta}(Q, R, I)$ and $\mathbf{Id} \Rightarrow G$, then $R, G, I \models_{ST} \{Q\}\mathbf{skip}\{Q\}$.*

Proof. For all Q, R, G, I such that $\mathbf{Sta}(Q, R, I)$ and $\mathbf{Id} \Rightarrow G$, by Def. 34 we need to prove for all μ, if $\mu \models I \wedge Q$, then for all n, $(init(\mathbf{skip}, \mu), R, I) \Longmapsto^{n}_{\mathrm{ST}} (G, Q)$. For all μ such that $\mu \models I \wedge Q$, by Lem. 18 we know $(\delta(\mathbf{skip}) \otimes \mu)^{(Stmt)} = \delta(\mathbf{skip})$. From $\mathbf{Sta}(Q, R, I)$ and $\mathbf{Id} \Rightarrow G$ by Lem. 194 we have $(init(\mathbf{skip}, \mu), R, I) \Longmapsto^{n}_{\mathrm{ST}} (G, Q)$ for all n.

Lemma 196. *For all η and b, $\eta^{(State)} \models \lceil b \rceil$ if and only if $\forall (C, \sigma) \in supp(\eta).\ \sigma \models b$.*

Proof. For all η and b, by Lem. 22 we know $\mathit{supp}(\eta^{(State)}) = \mathit{range}(\mathit{supp}(\eta))$, thus

$$\begin{aligned}
& \eta^{(State)} \models \lceil b \rceil \\
\iff & \forall \sigma \in \mathit{supp}(\eta^{(State)}).\ \sigma \models b \\
\iff & \forall \sigma \in \mathit{range}(\mathit{supp}(\eta)).\ \sigma \models b \\
\iff & \forall (C, \sigma) \in \mathit{supp}(\eta).\ \sigma \models b.
\end{aligned}$$

Lemma 197. *For all η, b, C_1, C_2, if $\eta^{(Stmt)} = \delta(\textbf{\textit{if}}\ (b)\ \textbf{\textit{then}}\ C_1\ \textbf{\textit{else}}\ C_2)$ and $\eta^{(State)} \models \lceil b \rceil$, then $\eta \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}, \delta(C_1) \otimes \eta^{(State)})$.*

Proof. For all η, b, C_1, C_2 such that $\eta^{(Stmt)} = \delta(\textbf{if}\ (b)\ \textbf{then}\ C_1\ \textbf{else}\ C_2)$ and $\eta^{(State)} \models \lceil b \rceil$, by Lem. 189 and Lem. 196 we know $C = \textbf{if}\ (b)\ \textbf{then}\ C_1\ \textbf{else}\ C_2$ and $\sigma \models b$ for all $(C, \sigma) \in \mathit{supp}(\eta)$, thus

$$\begin{aligned}
& \lambda(C', \sigma').\ \textstyle\sum_{C,\sigma} \{\eta(C, \sigma) \cdot p \mid (C, \sigma) \xrightarrow{p} (C', \sigma')\} \\
= {} & \lambda(C', \sigma').\ \textstyle\sum_{C,\sigma} \{\eta(C, \sigma) \cdot p \mid (C, \sigma) \in \mathit{supp}(\eta) \wedge (C, \sigma) \xrightarrow{p} (C', \sigma')\} \\
= {} & \lambda(C', \sigma').\ \textstyle\sum_{C,\sigma} \{\eta(C, \sigma) \cdot p \mid (C, \sigma) \in \mathit{supp}(\eta) \wedge C = \textbf{if}\ (b)\ \textbf{then}\ C_1\ \textbf{else}\ C_2 \wedge \\
& \qquad \sigma \models b \wedge (C, \sigma) \xrightarrow{p} (C', \sigma')\} \\
= {} & \lambda(C', \sigma').\ \textstyle\sum_{\sigma} \{\eta^{(State)}(\sigma) \cdot p \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma \models b \wedge \\
& \qquad (\textbf{if}\ (b)\ \textbf{then}\ C_1\ \textbf{else}\ C_2, \sigma) \xrightarrow{p} (C', \sigma')\} \\
= {} & \lambda(C', \sigma').\ \textstyle\sum_{\sigma} \{\eta^{(State)}(\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma \models b \wedge C' = C_1 \wedge \sigma' = \sigma\} \\
= {} & \lambda(C', \sigma').\ \textstyle\sum_{\sigma} \{\eta^{(State)}(\sigma) \mid C' = C_1 \wedge \sigma' = \sigma\} \\
= {} & \lambda(C', \sigma').\ \delta(C_1)(C') \cdot \eta^{(State)}(\sigma') \\
= {} & \delta(C_1) \otimes \eta^{(State)}.
\end{aligned}$$

and

$$\begin{aligned}
& \{(\sigma, \sigma') \mid \exists C, C'.\ \eta(C, \sigma) > 0 \wedge (C, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\
= {} & \{(\sigma, \sigma') \mid \exists C, C'.\ (C, \sigma) \in \mathit{supp}(\eta) \wedge (C, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\
= {} & \{(\sigma, \sigma') \mid \exists C, C'.\ (C, \sigma) \in \mathit{supp}(\eta) \wedge (C = \textbf{if}\ (b)\ \textbf{then}\ C_1\ \textbf{else}\ C_2) \wedge \\
& \qquad \sigma \models b \wedge (C, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\
= {} & \{(\sigma, \sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma \models b \wedge \exists C'.\ (\textbf{if}\ (b)\ \textbf{then}\ C_1\ \textbf{else}\ C_2, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\
= {} & \{(\sigma, \sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \exists C'.\ C' = C_1 \wedge \sigma' = \sigma\} \\
= {} & \{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}.
\end{aligned}$$

Therefore $\eta \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}, \delta(C_1) \otimes \eta^{(State)})$.

Lemma 198. *For all η, b, C_1, C_2, if $\eta^{(Stmt)} = \textbf{\textit{if}}\ (b)\ \textbf{\textit{then}}\ C_1\ \textbf{\textit{else}}\ C_2$ and $\eta^{(State)} \models \lceil \neg b \rceil$, then $\eta \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}, \delta(C_2) \otimes \eta^{(State)})$.*

Proof. For all η, b, C_1, C_2 such that $\eta^{(Stmt)} = \delta(\textbf{if}\ (b)\ \textbf{then}\ C_1\ \textbf{else}\ C_2)$ and $\eta^{(State)} \models \lceil \neg b \rceil$, by Lem. 189 and Lem. 196 we know $C = \textbf{if}\ (b)\ \textbf{then}\ C_1\ \textbf{else}\ C_2$

and $\sigma \models \neg b$ for all $(C,\sigma) \in \mathit{supp}(\eta)$, thus

$$
\begin{aligned}
&\lambda(C',\sigma').\textstyle\sum_{C,\sigma}\{\eta(C,\sigma)\cdot p \mid (C,\sigma) \xrightarrow{p} (C',\sigma')\}\\
=\ &\lambda(C',\sigma').\textstyle\sum_{C,\sigma}\{\eta(C,\sigma)\cdot p \mid (C,\sigma) \in \mathit{supp}(\eta) \wedge (C,\sigma) \xrightarrow{p} (C',\sigma')\}\\
=\ &\lambda(C',\sigma').\textstyle\sum_{C,\sigma}\{\eta(C,\sigma)\cdot p \mid (C,\sigma) \in \mathit{supp}(\eta) \wedge C = \textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2 \wedge\\
&\qquad \sigma \models b \wedge (C,\sigma) \xrightarrow{p} (C',\sigma')\}\\
=\ &\lambda(C',\sigma').\textstyle\sum_{\sigma}\{\eta^{(State)}(\sigma)\cdot p \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma \models \neg b \wedge\\
&\qquad (\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2,\sigma) \xrightarrow{p} (C',\sigma')\}\\
=\ &\lambda(C',\sigma').\textstyle\sum_{\sigma}\{\eta^{(State)}(\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma \models \neg b \wedge C' = C_1 \wedge \sigma' = \sigma\}\\
=\ &\lambda(C',\sigma').\textstyle\sum_{\sigma}\{\eta^{(State)}(\sigma) \mid C' = C_2 \wedge \sigma' = \sigma\}\\
=\ &\lambda(C',\sigma').\ \delta(C_2)(C')\cdot\eta^{(State)}(\sigma')\\
=\ &\delta(C_2)\otimes\eta^{(State)}
\end{aligned}
$$

and

$$
\begin{aligned}
&\{(\sigma,\sigma') \mid \exists C,C'.\ \eta(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
=\ &\{(\sigma,\sigma') \mid \exists C,C'.\ (C,\sigma) \in \mathit{supp}(\eta) \wedge (C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
=\ &\{(\sigma,\sigma') \mid \exists C,C'.\ (C,\sigma) \in \mathit{supp}(\eta) \wedge (C = \textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2) \wedge\\
&\qquad \sigma \models b \wedge (C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
=\ &\{(\sigma,\sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma \models \neg b \wedge \exists C'.\ (\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
=\ &\{(\sigma,\sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \exists C'.\ C' = C_2 \wedge \sigma' = \sigma\}\\
=\ &\{(\sigma,\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}.
\end{aligned}
$$

Therefore $\eta \rightsquigarrow (\{(\sigma,\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}, \delta(C_2)\otimes\eta^{(State)})$.

Lemma 199. *For all W, if $W^{(Prog)}(\boldsymbol{skip} \parallel \cdots \parallel \boldsymbol{skip}) = 1$, then $W|_{\boldsymbol{skip}\parallel\cdots\parallel\boldsymbol{skip}} = W$.*

Proof. For all η such that $\eta^{(Stmt)}(\mathbf{skip}) = 1$, we have $\mathbf{Pr}_{(C,\sigma)\sim\eta}[\mathbb{C} = \mathbf{skip}] = 1$. By Lem. 4 we know $\eta|_{\lambda(C,\sigma).C=\mathbf{skip}} = \eta$, i.e., $\eta|_{\mathbf{skip}} = \eta$.

Lemma 200 (Soundness of (COND) rule). *For all $b, C_1, C_2, R, G, I, P_1, P_2, Q$, if $\mathbf{Sta}(P_1\vee P_2, R, I)$, $P_1 \Rightarrow \lceil b\rceil$, $P_2 \Rightarrow \lceil\neg b\rceil$, $\mathbf{Id} \Rightarrow G$, $R,G,I \models_\square \{P_1\}C_1\{Q\}$ and $R,G,I \models_\square \{P_2\}C_2\{Q\}$, then $R,G,I \models_\square \{P_1 \vee P_2\}\boldsymbol{if}\ (b)\ \boldsymbol{then}\ C_1\ \boldsymbol{else}\ C_2\{Q\}$.*

Proof. For all $b, C_1, C_2, R, G, I, P_1, P_2, Q$ such that $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $P_1 \Rightarrow \lceil b\rceil$, $P_2 \Rightarrow \lceil\neg b\rceil$, $\mathbf{Id} \Rightarrow G$, $R,G,I \models_\square \{P_1\}C_1\{Q\}$ and $R,G,I \models_\square \{P_2\}C_2\{Q\}$, we need to prove for all μ, if $\mu \models I\wedge(P_1\vee P_2)$, then $(\mathit{init}(\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2,\mu), R, I) \Longrightarrow^n_{\mathrm{ST}} (G,Q)$ for all n. For all μ such that $\mu \models I \wedge (P_1 \vee P_2)$, by Lem. 18 we know $\mathit{init}(\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2,\mu)^{(Stmt)} = (\delta(\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2)\otimes\mu)^{(Stmt)} = \delta(\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2)$. To prove $(\mathit{init}(\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2,\mu), R, I) \Longrightarrow^n_{\mathrm{ST}} (G,Q)$ for all n, it suffices to prove for all n and η, if $\eta^{(Stmt)} = \delta(\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2)$ and $\eta^{(State)} \models I \wedge (P_1 \vee P_2)$, then $(\eta, R, I) \Longrightarrow^n_{\mathrm{ST}} (G,Q)$. We prove by induction on n.

– base case: $n = 0$. trivial.

– inductive case: $n = k + 1$.
IH: for all η, if ${\eta^{(Stmt)}} = \delta(\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2)$ and ${\eta^{(State)}} \models I \wedge (P_1 \vee P_2)$, then $(\eta, R, I) \Longmapsto^{k}_{\mathrm{ST}} (G, Q)$.
For all η such that ${\eta^{(Stmt)}} = \delta(\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2)$ and ${\eta^{(State)}} \models I \wedge (P_1 \vee P_2)$, to prove $(\eta, R, I) \Longmapsto^{k+1}_{\mathrm{ST}} (G, Q)$, we need to prove
  - ${\eta^{(Stmt)}}(\textbf{skip}) = 0$ or ${\eta^{(Stmt)}}(\textbf{skip}) = 1$.
    From ${\eta^{(Stmt)}} = \delta(\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2)$ we have ${\eta^{(Stmt)}}(\textbf{skip}) = \delta(\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2)(\textbf{skip}) = 0$.
  - if ${\eta^{(Stmt)}}(\textbf{skip}) > 0$, then ${\eta|_{\textbf{skip}}}^{(State)} \models Q$.
    ${\eta^{(Stmt)}}(\textbf{skip}) > 0$ contradicts with ${\eta^{(Stmt)}}(\textbf{skip}) = 0$.
  - ${\eta^{(State)}} \models I$.
    From ${\eta^{(State)}} \models I \wedge (P_1 \vee P_2)$ we have ${\eta^{(State)}} \models I$.
  - for all η', if $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longmapsto^{k}_{\mathrm{ST}} (G, Q)$.
    For all η' such that $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, by Lem. 188 we know $\mathit{supp}({\eta'}^{(Stmt)}) \subseteq \mathit{supp}({\eta}^{(Stmt)})$. From ${\eta^{(Stmt)}} = \delta(\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2)$ by Lem. 27 we know ${\eta'}^{(Stmt)} = \delta(\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2)$. From $\mathbf{Sta}(P_1 \vee P_2, R, I)$, ${\eta^{(State)}} \models I \wedge (P_1 \vee P_2)$ and $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$ by Lem. 186 we have ${\eta'}^{(State)} \models I \wedge (P_1 \vee P_2)$. From ${\eta'}^{(Stmt)} = \delta(\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2)$ by IH we have $(\eta', R, I) \Longmapsto^{k}_{\mathrm{ST}} (G, Q)$.
  - for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, ${\eta'}^{(State)} \models I$ and $(\eta', R, I) \Longmapsto^{k}_{\mathrm{ST}} (G, Q)$.
    For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, from ${\eta^{(Stmt)}} = \delta(\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2)$ by Lem. 190 we have $\mathit{nextsplit}(\eta) = \{\mathit{nextsplit}(\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2)\} = \{\textbf{split}(\mathrm{true})\}$. From $\eta \hookrightarrow (\theta, \eta')$ by Lem. 191 we know $\eta \rightsquigarrow (\theta, \eta')$. From ${\eta^{(State)}} \models I \wedge (P_1 \vee P_2)$ we know ${\eta^{(State)}} \models I \wedge P_1$ or ${\eta^{(State)}} \models I \wedge P_2$. We prove the two cases respectively.
    * case 1: ${\eta^{(State)}} \models I \wedge P_1$.
      From $P_1 \Rightarrow \lceil b \rceil$ we know ${\eta^{(State)}} \models \lceil b \rceil$. By Lem. 197 we know $\eta \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}({\eta^{(State)}})\}, \delta(C_1) \otimes {\eta^{(State)}})$. From $\eta \rightsquigarrow (\theta, \eta')$ by Lem. 193 we have $\theta = \{(\sigma, \sigma) \mid \sigma \in \mathit{supp}({\eta^{(State)}})\}$ and $\eta' = \delta(C_1) \otimes {\eta^{(State)}}$, thus $\theta \subseteq [\![\mathbf{Id}]\!] \subseteq [\![G]\!]$ and ${\eta'}^{(State)} = {\eta^{(State)}} \models I$. From $R, G, I \models_{\square} \{P_1\} C_1 \{Q\}$, $\eta' = \delta(C_1) \otimes {\eta^{(State)}} = \mathit{init}(C_1, {\eta^{(State)}})$ and ${\eta^{(State)}} \models I \wedge P_1$ we have $(\eta', R, I) \Longmapsto^{k}_{\square} (G, Q)$.
    * case 2: ${\eta^{(State)}} \models I \wedge P_2$.
      From $P_2 \Rightarrow \lceil \neg b \rceil$ we know ${\eta^{(State)}} \models \lceil \neg b \rceil$. By Lem. 198 we know $\eta \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}({\eta^{(State)}})\}, \delta(C_2) \otimes {\eta^{(State)}})$. From $\eta \rightsquigarrow (\theta, \eta')$ by Lem. 193 we have $\theta = \{(\sigma, \sigma) \mid \sigma \in \mathit{supp}({\eta^{(State)}})\}$ and $\eta' = \delta(C_2) \otimes {\eta^{(State)}}$, thus $\theta \subseteq [\![\mathbf{Id}]\!] \subseteq [\![G]\!]$ and ${\eta'}^{(State)} = {\eta^{(State)}} \models I$. From $R, G, I \models_{\square} \{P_2\} C_2 \{Q\}$, $\eta' = \delta(C_2) \otimes {\eta^{(State)}} = \mathit{init}(C_2, {\eta^{(State)}})$ and ${\eta^{(State)}} \models I \wedge P_2$ we have $(\eta', R, I) \Longmapsto^{k}_{\square} (G, Q)$.

Lemma 201. *For all* η, C_2, ${\eta; C_2}^{(State)} = {\eta^{(State)}}$.

Proof. For all η, C_2,

$$\begin{aligned}\eta; {C_2}^{(State)} &= \lambda\sigma. \sum_C \eta; C_2(C, \sigma) \\ &= \lambda\sigma. \sum_{C_1} \eta; C_2(C_1; C_2, \sigma) \\ &= \lambda\sigma. \sum_{C_1} \eta(C_1, \sigma) \\ &= \eta^{(State)}.\end{aligned}$$

Lemma 202. *For all η, C_2, η', if $\eta' = \lambda(C, \sigma).\ \eta(C; C_2, \sigma)$ and for all $C \in supp(\eta^{(Stmt)})$, there exists C_1 such that $C = C_1; C_2$, then $\eta'; C_2 = \eta$.*

Proof. For all η, C_2, η' such that $\eta' = \lambda(C, \sigma).\ \eta(C; C_2, \sigma)$ and for all $C \in supp(\eta^{(Stmt)})$, there exists C_1 such that $C = C_1; C_2$, we have for all C, if there is no C_1 such that $C = C_1; C_2$, then $C \notin supp(\eta^{(Stmt)})$, i.e., $\eta(C, \sigma) = 0$ for all σ.

$$\begin{aligned}\eta'; C_2 &= \lambda(C, \sigma). \begin{cases} \eta'(C_1, \sigma), & \text{if } C = C_1; C_2 \\ 0, & \text{otherwise} \end{cases} \\ &= \lambda(C, \sigma). \begin{cases} \eta(C_1; C_2, \sigma), & \text{if } C = C_1; C_2 \\ 0, & \text{otherwise} \end{cases} \\ &= \lambda(C, \sigma). \begin{cases} \eta(C, \sigma), & \text{if } C = C_1; C_2 \\ \eta(C, \sigma), & \text{otherwise} \end{cases} \\ &= \eta.\end{aligned}$$

Lemma 203. *For all R, η, C_2, η', if $\eta; C_2 \overset{R}{\mapsto} \eta'$, then there exists η'' such that $\eta \overset{R}{\mapsto} \eta''$ and $\eta' = \eta''; C_2$.*

Proof. For all R, η, C_2, η' such that $\eta; C_2 \overset{R}{\mapsto} \eta'$, there exists ψ such that $dom(\psi) = supp(\eta; C_2)$, $range(\psi) = supp(\eta')$ and for all $((C, \sigma), (C', \sigma')) \in \psi$, $C' = C$ and $(\sigma, \sigma') \models R$. From $\eta; C_2 \overset{R}{\mapsto} \eta'$ by Lem. 187 we know $supp({\eta'}^{(Stmt)}) = supp(\eta; {C_2}^{(Stmt)})$, thus for all $C \in {\eta'}^{(Stmt)}$, we have $\eta; {C_2}^{(Stmt)} > 0$, so there exists C_1 such that $C = C_1; C_2$. Let $\eta'' \overset{\text{def}}{=} \lambda(C, \sigma).\ \eta'(C; C_2, \sigma)$, by Lem. 202 we know $\eta' = \eta''; C_2$. For all $((C, \sigma), (C', \sigma')) \in \psi$, we have $C' = C$ and $(C, \sigma) \in dom(\psi) = supp(\eta; C_2)$, so there exists C_1 such that $C' = C = C_1; C_2$. Let $\psi' \overset{\text{def}}{=} \{((C, \sigma), (C', \sigma')) \mid ((C; C_2, \sigma), (C'; C_2, \sigma')) \in \psi\}$, we have

$$\begin{aligned}dom(\psi') &= \{(C, \sigma) \mid \exists C', \sigma'.\ ((C, \sigma), (C', \sigma')) \in \psi'\} \\ &= \{(C, \sigma) \mid \exists C', \sigma'.\ ((C; C_2, \sigma), (C'; C_2, \sigma')) \in \psi\} \\ &= \{(C, \sigma) \mid \exists C', \sigma'.\ ((C; C_2, \sigma), (C', \sigma')) \in \psi\} \\ &= \{(C, \sigma) \mid (C; C_2, \sigma) \in dom(\psi)\} \\ &= \{(C, \sigma) \mid (C; C_2, \sigma) \in supp(\eta)\} \\ &= \{(C, \sigma) \mid (\eta; C_2)(C; C_2, \sigma) > 0\} \\ &= \{(C, \sigma) \mid \eta(C, \sigma) > 0\} \\ &= supp(\eta),\end{aligned}$$

$$\begin{aligned}
range(\psi') &= \{(C',\sigma') \mid \exists C,\sigma.\ ((C,\sigma),(C',\sigma')) \in \psi'\} \\
&= \{(C',\sigma') \mid \exists C,\sigma.\ ((C;C_2,\sigma),(C';C_2,\sigma')) \in \psi\} \\
&= \{(C',\sigma') \mid \exists C,\sigma.\ ((C,\sigma),(C';C_2,\sigma')) \in \psi\} \\
&= \{(C',\sigma') \mid (C';C_2,\sigma') \in dom(\psi)\} \\
&= \{(C',\sigma') \mid (C';C_2,\sigma') \in supp(\eta')\} \\
&= \{(C',\sigma') \mid \eta'(C';C_2,\sigma') > 0\} \\
&= \{(C',\sigma') \mid \eta''(C',\sigma') > 0\} \\
&= supp(\eta''),
\end{aligned}$$

and for all $((C,\sigma),(C',\sigma')) \in \psi'$, we have $((C;C_2,\sigma),(C';C_2,\sigma')) \in \psi$, so $C';C_2 = C;C_2$ and $(\sigma,\sigma') \models R$, thus $C' = C$ and $(\sigma,\sigma') \models R$.

Lemma 204. *For all η and b, $[\![\boldsymbol{Pr}(b)]\!]_{\eta^{(State)}} = \boldsymbol{Pr}_{(C,\sigma)\sim\eta}[\sigma \models b]$.*

Proof. For all η and b, by Lem. 3 we know $[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}} = \mathbf{Pr}_{\sigma\sim\eta^{(State)}}[\sigma \models b] = \mathbf{Pr}_{(C,\sigma)\sim\eta}[\sigma \models b]$.

Lemma 205. *For all η and b, $\eta|_b$ exists if and only if $[\![\boldsymbol{Pr}(b)]\!]_{\eta^{(State)}} > 0$.*

Proof. For all η and b, by definition of $\eta|_b$ we know $\eta|_b$ exists if and only if $\eta|_{\lambda(C,\sigma).\ \sigma\models b}$ exists. By Def. 2 we know $\eta|_{\lambda(C,\sigma).\ \sigma\models b}$ exists if and only if $\mathbf{Pr}_{(C,\sigma)\sim W}[\sigma \models b] > 0$. By Lem. 204 we know $\mathbf{Pr}_{(C,\sigma)\sim\eta}[\sigma \models b] > 0$ if and only if $[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}} > 0$. Therefore, $\eta|_b$ exists if and only if $[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}} > 0$.

Lemma 206. *For all η and b, if $[\![\boldsymbol{Pr}(b)]\!]_{\eta^{(State)}} > 0$, then $\eta^{(State)}|_b = \eta|_b{}^{(State)}$.*

Proof. For all η and b, if $[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}} > 0$,

$$\begin{aligned}
&\eta^{(State)}|_b \\
&= \lambda\sigma. \begin{cases} \frac{\eta^{(State)}(\sigma)}{[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}}, & \text{if } \sigma \models b \\ 0, & \text{otherwise} \end{cases} \\
&= \lambda\sigma. \begin{cases} \frac{\sum_C \eta(C,\sigma)}{[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}}, & \text{if } \sigma \models b \\ 0, & \text{otherwise} \end{cases} \\
&= \lambda\sigma. \sum_C \begin{cases} \frac{\eta(C,\sigma)}{[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}}, & \text{if } \sigma \models b \\ 0, & \text{otherwise} \end{cases} \\
&= \lambda\sigma. \sum_C \eta|_b(C,\sigma) \\
&= \eta|_b{}^{(State)}.
\end{aligned}$$

Lemma 207. *For all η and b, if $[\![\boldsymbol{Pr}(b)]\!]_{\eta^{(State)}} > 0$, then $\eta|_b = \lambda(C,\sigma).\ \frac{\chi(\sigma\models b_i)\cdot\eta(C,\sigma)}{[\![\boldsymbol{Pr}(b)]\!]_{\eta^{(State)}}}$.*

Proof. For all η, b such that $[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}} > 0$, we have $\mathbf{Pr}_{(C,\sigma)\sim W}[\sigma \models b] > 0$, thus

$$\begin{aligned}
\eta|_b &= \eta|_{\lambda(C,\sigma).\sigma\models b} \\
&= \lambda(C,\sigma). \begin{cases} \frac{W(C,\sigma)}{\mathbf{Pr}_{(C,\sigma)\sim W}[\sigma\models b]}, & \text{if } \sigma \models b \\ 0, & \text{otherwise} \end{cases} \\
&= \lambda(C,\sigma).\ \frac{\chi(\sigma\models b)\cdot W(C,\sigma)}{\mathbf{Pr}_{(C,\sigma)\sim W}[\sigma\models b]} \\
&= \lambda(C,\sigma).\ \frac{\chi(\sigma\models b)\cdot W(C,\sigma)}{[\![\mathbf{Pr}(b)]\!]_{W^{(State)}}}. \qquad \text{(by Lem. 204)}
\end{aligned}$$

Lemma 208. *For all η, b, C_2, if $[\![\boldsymbol{Pr}(b)]\!]_{\eta^{(State)}} > 0$, then $(\eta; C_2)|_b = \eta|_b; C_2$.*

Proof. For all η, b, C_2 such that $[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}} > 0$, by Lem. 201 we know ${\eta; C_2}^{(State)} = \eta^{(State)}$, thus $[\![\mathbf{Pr}(b)]\!]_{\eta; {C_2}^{(State)}} > 0$. By Lem. 205 we know both $(\eta; C_2)|_b$ and $\eta|_b$ exists.

$$
\begin{aligned}
& \eta|_b; C_2 \\
= {} & \lambda(C, \sigma). \begin{cases} \eta|_b(C_1, \sigma), & \text{if } C = C_1; C_2 \\ 0, & \text{otherwise} \end{cases} \\
= {} & \lambda(C, \sigma). \begin{cases} \frac{\chi(\sigma \models b) \cdot \eta(C_1, \sigma)}{[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}}, & \text{if } C = C_1; C_2 \\ 0, & \text{otherwise} \end{cases} \qquad \text{(by Lem. 207)} \\
= {} & \lambda(C, \sigma). \begin{cases} \frac{\eta(C_1, \sigma)}{[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}}, & \text{if } C = C_1; C_2 \wedge \sigma \models b \\ 0, & \text{otherwise} \end{cases} \\
= {} & \lambda(C, \sigma). \begin{cases} \frac{\eta; C_2(C_1; C_2, \sigma)}{[\![\mathbf{Pr}(b)]\!]_{\eta; {C_2}^{(State)}}}, & \text{if } C = C_1; C_2 \wedge \sigma \models b \\ 0, & \text{otherwise} \end{cases} \\
= {} & \lambda(C, \sigma). \begin{cases} \frac{\eta; C_2(C, \sigma)}{[\![\mathbf{Pr}(b)]\!]_{\eta; {C_2}^{(State)}}}, & \text{if } \sigma \models b \\ 0, & \text{otherwise} \end{cases} \\
= {} & (\eta; C_2)|_b.
\end{aligned}
$$

Lemma 209. *For all R, I, η_1, C_2, η', if $\eta_1; C_2 \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then there exists η_1' such that $\eta_1 \overset{R}{\underset{I}{\rightarrowtail}} \eta_1'$ and $\eta' = \eta_1'; C_2$.*

Proof. For all R, I, η_1, C_2, η' such that $\eta_1; C_2 \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, there exists η'' and b such that $\eta_1; C_2 \overset{R}{\mapsto} \eta''$, $\eta''|_b = \eta'$, ${\eta'}^{(State)} \models I$. From $\eta_1; C_2 \overset{R}{\mapsto} \eta''$ by Lem. 203 there exists η_1'' such that $\eta \overset{R}{\mapsto} \eta_1''$ and $\eta'' = \eta_1''; C_2$. From $\eta''|_b = \eta'$ by Lem. 205 we know $[\![\mathbf{Pr}(b)]\!]_{{\eta''}^{(State)}} > 0$. By Lem. 201 we know ${\eta_1''}^{(State)} = {\eta_1''; C_2}^{(State)} = {\eta''}^{(State)}$, thus $[\![\mathbf{Pr}(b)]\!]_{{\eta_1''}^{(State)}} > 0$. By Lem. 208 we know $\eta' = \eta''|_b = (\eta_1''; C_2)|_b = \eta_1''|_b; C_2$. Let $\eta_1' \overset{\text{def}}{=} \eta_1''|_b$, then $\eta' = \eta_1'; C_2$. From $\eta \overset{R}{\mapsto} \eta_1''$ and $\eta_1''|_b = \eta_1'$ we know $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta_1'$.

Lemma 210. *For all η, C_2, θ, η', if ${\eta}^{(Stmt)} = \delta(\boldsymbol{skip})$, then $\eta; C_2 \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in supp({\eta}^{(State)})\}, \delta(C_2) \otimes {\eta}^{(State)})$.*

Proof. For all η such that $\eta^{(Stmt)} = \delta(\mathbf{skip})$, by Lem. 189 we know $C = \mathbf{skip}$ for all $(C,\sigma) \in supp(\eta)$, thus

$$
\begin{aligned}
&\lambda(C',\sigma').\textstyle\sum_{C,\sigma}\{\eta;C_2(C,\sigma)\cdot p \mid (C,\sigma) \xrightarrow{p} (C',\sigma')\}\\
=\ &\lambda(C',\sigma').\textstyle\sum_{C_1,\sigma}\{\eta(C_1,\sigma)\cdot p \mid (C_1,\sigma)\in supp(\eta)\wedge (C_1;C_2,\sigma)\xrightarrow{p}(C',\sigma')\}\\
=\ &\lambda(C',\sigma').\textstyle\sum_{C_1,\sigma}\{\eta(C_1,\sigma)\cdot p \mid (C_1,\sigma)\in supp(\eta)\wedge C_1=\mathbf{skip}\wedge (C_1;C_2,\sigma)\xrightarrow{p}(C',\sigma')\}\\
=\ &\lambda(C',\sigma').\textstyle\sum_{\sigma}\{\eta^{(State)}(\sigma)\cdot p \mid \sigma\in supp(\eta^{(State)})\wedge (\mathbf{skip};C_2,\sigma)\xrightarrow{p}(C',\sigma')\}\\
=\ &\lambda(C',\sigma').\textstyle\sum_{\sigma}\{\eta^{(State)}(\sigma) \mid \sigma\in supp(\eta^{(State)})\wedge C'=C_2\wedge\sigma'=\sigma\}\\
=\ &\lambda(C',\sigma').\textstyle\sum_{\sigma}\{\eta^{(State)}(\sigma) \mid C'=C_2\wedge\sigma'=\sigma\}\\
=\ &\lambda(C',\sigma').\ \delta(C_2)(C')\cdot\eta^{(State)}(\sigma')\\
=\ &\delta(C_2)\otimes\eta^{(State)}
\end{aligned}
$$

and

$$
\begin{aligned}
&\{(\sigma,\sigma') \mid \exists C,C'.\ \eta;C_2(C,\sigma)>0\wedge(C,\sigma)\xrightarrow{p}(C',\sigma')\wedge p>0\}\\
=\ &\{(\sigma,\sigma') \mid \exists C_1,C'.\ (C_1,\sigma)\in supp(\eta)\wedge(C_1;C_2,\sigma)\xrightarrow{p}(C',\sigma')\wedge p>0\}\\
=\ &\{(\sigma,\sigma') \mid \exists C_1,C'.\ (C_1,\sigma)\in supp(\eta)\wedge(C_1=\mathbf{skip})\wedge(C_1;C_2,\sigma)\xrightarrow{p}(C',\sigma')\wedge p>0\}\\
=\ &\{(\sigma,\sigma') \mid \sigma\in supp(\eta^{(State)})\wedge\exists C'.\ (\mathbf{skip};C_2,\sigma)\xrightarrow{p}(C',\sigma')\wedge p>0\}\\
=\ &\{(\sigma,\sigma') \mid \sigma\in supp(\eta^{(State)})\wedge\exists C'.\ C'=C_2\wedge\sigma'=\sigma\}\\
=\ &\{(\sigma,\sigma) \mid \sigma\in supp(\eta^{(State)})\}.
\end{aligned}
$$

Therefore $\eta \rightsquigarrow (\{(\sigma,\sigma) \mid \sigma\in supp(\eta^{(State)})\}, \delta(C_2)\otimes\eta^{(State)})$.

Lemma 211. *For all R,G,I,Q,n,η, if $(\eta,R,I) \Longrightarrow_{\square}^{n+1} (G,Q)$, then $(\eta,R,I) \Longrightarrow_{\square}^{n} (G,Q)$.*

Proof. For all R,G,I,Q, we prove for all n,η, if $(\eta,R,I) \Longrightarrow_{\square}^{n+1} (G,Q)$, then $(\eta,R,I) \Longrightarrow_{\square}^{n} (G,Q)$ by induction on n.

- base case: $n=0$. trivial.
- inductive case: $n=k+1$.
  IH: for all η, if $(\eta,R,I) \Longrightarrow_{\square}^{k+1} (G,Q)$, then $(\eta,R,I) \Longrightarrow_{\square}^{k} (G,Q)$.
  For all η such that $(\eta,R,I) \Longrightarrow_{\square}^{n+1} (G,Q)$, to prove $(\eta,R,I) \Longrightarrow_{\square}^{n} (G,Q)$, i.e., $(\eta,R,I) \Longrightarrow_{\square}^{k+1} (G,Q)$, we need to prove
  - (when $\square = \text{ST}$) $\eta^{(Stmt)}(\mathbf{skip}) = 1$ or $\eta^{(Stmt)}(\mathbf{skip}) = 0$.
    From $(\eta,R,I) \Longrightarrow_{\square}^{n+1} (G,Q)$ we know $\eta^{(Stmt)}(\mathbf{skip}) = 1$ or $\eta^{(Stmt)}(\mathbf{skip}) = 0$.
  - if $\eta^{(Stmt)}(\mathbf{skip}) > 0$, then $\eta|_{\mathbf{skip}}{}^{(State)} \models Q$.
    From $(\eta,R,I) \Longrightarrow_{\square}^{n+1} (G,Q)$ we know if $\eta^{(Stmt)}(\mathbf{skip}) > 0$, then $\eta|_{\mathbf{skip}}{}^{(State)} \models Q$.
  - $\eta^{(State)} \models I$.
    From $(\eta,R,I) \Longrightarrow_{\square}^{n+1} (G,Q)$ we know $\eta^{(State)} \models I$.
  - for all η', if $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $(\eta',R,I) \Longrightarrow_{\square}^{k} (G,Q)$.
    For all η' such that $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, from $(\eta,R,I) \Longrightarrow_{\square}^{n+1} (G,Q)$ we know $(\eta',R,I) \Longrightarrow_{\square}^{n} (G,Q)$, i.e., $(\eta',R,I) \Longrightarrow_{\square}^{k+1} (G,Q)$. By IH we have $(\eta',R,I) \Longrightarrow_{\square}^{k} (G,Q)$.

- for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, ${\eta'}^{(State)} \models I$ and $(\eta', R, I) \Longmapsto_{\square}^{k} (G, Q)$.
  For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, from $(\eta, R, I) \Longmapsto_{\square}^{n+1} (G, Q)$ we know $\theta \subseteq [\![G]\!]$, ${\eta'}^{(State)} \models I$ and $(\eta', R, I) \Longmapsto_{\square}^{n} (G, Q)$, thus $(\eta', R, I) \Longmapsto_{\square}^{k+1} (G, Q)$. By IH we have $(\eta', R, I) \Longmapsto_{\square}^{k} (G, Q)$.

Lemma 212. *For all η and C_1, $\eta^{(Stmt)}(C_1) = 0$ if and only if $\forall (C, \sigma) \in supp(\eta).\ C \neq C_1$.*

Proof. For all η and C_1, we have

$$\begin{array}{ll} & \eta^{(Stmt)}(C_1) = 0 \\ \Longleftrightarrow & C_1 \notin supp(\eta^{(Stmt)}) \\ \Longleftrightarrow & C_1 \notin dom(supp(\eta)) \qquad \text{(by Lem. 21)} \\ \Longleftrightarrow & C_1 \notin \{C \mid \exists \sigma.\ (C, \sigma) \in supp(\eta)\} \\ \Longleftrightarrow & \forall (C, \sigma) \in supp(\eta).\ C \neq C_1. \end{array}$$

Lemma 213. *For all $\eta_1, C_2, \theta, \eta'$, if ${\eta_1}^{(Stmt)}(\textbf{\textit{skip}}) = 0$ and $\eta_1; C_2 \rightsquigarrow (\theta, \eta')$, then there exists η_1' such that $\eta' = \eta_1'; C_2$ and $\eta_1 \rightsquigarrow (\theta, \eta_1')$.*

Proof. For all $\eta_1, C_2, \theta, \eta'$ such that $\eta^{(Stmt)}(\textbf{skip}) = 0$ and $\eta_1; C_2 \rightsquigarrow (\theta, \eta')$, from $\eta^{(Stmt)}(\textbf{skip}) = 0$ by Lem. 212 we know $\forall (C, \sigma) \in supp(\eta).\ C \neq C_1$. Let $\eta_1' \stackrel{\text{def}}{=} \lambda(C', \sigma').\sum_{C,\sigma}\{\eta_1(C, \sigma) \cdot p \mid (C, \sigma) \xrightarrow{p} (C', \sigma')\}$. From $\eta_1; C_2 \rightsquigarrow (\theta, \eta')$ we have

$$\begin{array}{ll} \eta' & = \lambda(C', \sigma').\sum_{C,\sigma}\{\eta_1; C_2(C, \sigma) \cdot p \mid (C, \sigma) \xrightarrow{p} (C', \sigma')\} \\ & = \lambda(C', \sigma').\sum_{C_1,\sigma}\{\eta_1(C_1, \sigma) \cdot p \mid (C_1, \sigma) \in supp(\eta) \wedge (C_1; C_2, \sigma) \xrightarrow{p} (C', \sigma')\} \\ & = \lambda(C', \sigma').\sum_{C_1,\sigma}\{\eta_1(C_1, \sigma) \cdot p \mid (C_1, \sigma) \in supp(\eta) \wedge C_1 \neq \textbf{skip} \wedge (C_1; C_2, \sigma) \xrightarrow{p} (C', \sigma')\} \\ & = \lambda(C', \sigma').\begin{cases} \sum_{C_1,\sigma}\{\eta_1(C_1, \sigma) \cdot p \mid (C_1, \sigma) \xrightarrow{p} (C_1', \sigma')\}, & \text{if } C' = C_1'; C_2 \\ 0, & \text{otherwise} \end{cases} \\ & = \lambda(C', \sigma').\begin{cases} \eta_1'(C_1', \sigma'), & \text{if } C' = C_1'; C_2 \\ 0, & \text{otherwise} \end{cases} \\ & = \eta_1'; C_2 \end{array}$$

and

$$\begin{array}{ll} \theta & = \{(\sigma, \sigma') \mid \exists C, C'.\ \eta_1; C_2(C, \sigma) > 0 \wedge (C, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\ & = \{(\sigma, \sigma') \mid \exists C_1, C'.\ \eta_1(C_1, \sigma) > 0 \wedge (C_1; C_2, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\ & = \{(\sigma, \sigma') \mid \exists C_1, C'.\ (C_1, \sigma) \in supp(\eta_1) \wedge (C_1; C_2, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\ & = \{(\sigma, \sigma') \mid \exists C_1, C'.\ (C_1, \sigma) \in supp(\eta_1) \wedge C_1 \neq \textbf{skip} \wedge (C_1; C_2, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\ & = \{(\sigma, \sigma') \mid \exists C_1, C_1'.\ \eta_1(C_1, \sigma) > 0 \wedge (C_1, \sigma) \xrightarrow{p} (C_1', \sigma') \wedge p > 0\}, \end{array}$$

thus $\eta_1 \rightsquigarrow (\theta, \eta_1')$.

Lemma 214. *For all η and C_2, $nextsplit(\eta; C_2) = nextsplit(\eta)$.*

Proof. For all η and C_2, we have

$$\begin{aligned} \mathit{nextsplit}(\eta; C_2) &= \{\mathit{nextsplit}(C) \mid \exists \sigma.\ (C, \sigma) \in \mathit{supp}(\eta; C_2)\} \\ &= \{\mathit{nextsplit}(C) \mid \exists \sigma.\ (\eta; C_2)(C, \sigma) > 0\} \\ &= \{\mathit{nextsplit}(C) \mid \exists \sigma, C_1.\ C = C_1; C_2 \wedge \eta(C_1, \sigma) > 0\} \\ &= \{\mathit{nextsplit}(C_1; C_2) \mid \exists \sigma.\ \eta(C_1, \sigma) > 0\} \\ &= \{\mathit{nextsplit}(C_1; C_2) \mid \exists \sigma.\ (C_1, \sigma) \in \mathit{supp}(\eta)\} \\ &= \mathit{nextsplit}(\eta). \end{aligned}$$

Lemma 215. *For all $\eta_1, C_2, \theta, \eta'$, if ${\eta_1}^{(\mathit{Stmt})}(\boldsymbol{skip}) = 0$ and $\eta_1; C_2 \hookrightarrow (\theta, \eta')$, then there exists η_1' such that $\eta' = \eta_1'; C_2$ and $\eta_1 \hookrightarrow (\theta, \eta_1')$.*

Proof. For all $\eta_1, C_2, \theta, \eta'$ such that $\eta^{(\mathit{Stmt})}(\mathbf{skip}) = 0$ and $\eta_1; C_2 \hookrightarrow (\theta, \eta')$, there are two cases.

- there exists $\eta'', b_1, \ldots, b_k, i$ such that $\eta_1; C_2 \rightsquigarrow (\theta, \eta'')$, $\mathit{nextsplit}(\eta_1; C_2) = \mathbf{split}(b_1, \ldots, b_k)$ and $\eta''|_{b_i} = \eta'$.
  From ${\eta_1}^{(\mathit{Stmt})}(\mathbf{skip}) = 0$ and $\eta_1; C_2 \rightsquigarrow (\theta, \eta'')$ by Lem. 213 we know there exists η_1'' such that $\eta'' = \eta_1''; C_2$ and $\eta_1 \rightsquigarrow (\theta, \eta_1'')$. From $\eta''|_{b_i} = \eta'$ by Lem. 205 we know $[\![\mathbf{Pr}(b_i)]\!]_{{\eta''}^{(\mathit{State})}} > 0$. By Lem. 201 we know ${\eta_1''; C_2}^{(\mathit{State})} = {\eta_1''}^{(\mathit{State})}$, thus $[\![\mathbf{Pr}(b_i)]\!]_{{\eta_1''}^{(\mathit{State})}} = [\![\mathbf{Pr}(b_i)]\!]_{{\eta_1'';C_2}^{(\mathit{State})}} = [\![\mathbf{Pr}(b_i)]\!]_{{\eta''}^{(\mathit{State})}} > 0$.
  By Lem. 205 we know $\eta_1''|_{b_i}$ exists. Let $\eta_1' \stackrel{\text{def}}{=} \eta_1''|_{b_i}$, from $[\![\mathbf{Pr}(b_i)]\!]_{{\eta_1''}^{(\mathit{State})}} > 0$ by Lem. 208 we know $\eta_1'; C_2 = \eta_1''|_{b_i}; C_2 = (\eta_1''; C_2)|_{b_i} = \eta''|_{b_i} = \eta'$. By Lem. 214 we know $\mathit{nextsplit}(\eta_1) = \mathit{nextsplit}(\eta_1; C_2) = \mathbf{split}(b_1, \ldots, b_k)$. From $\eta_1 \rightsquigarrow (\theta, \eta_1'')$ and $\eta_1''|_{b_i} = \eta_1'$ we know $\eta_1 \hookrightarrow (\theta, \eta_1'')$.
- $\eta_1; C_2 \rightsquigarrow (\theta, \eta')$ and $\#\mathit{nextsplit}(\eta_1; C_2) > 1$.
  From ${\eta_1}^{(\mathit{Stmt})}(\mathbf{skip}) = 0$ and $\eta_1; C_2 \rightsquigarrow (\theta, \eta')$ by Lem. 213 we know there exists η_1' such that $\eta' = \eta_1'; C_2$ and $\eta_1 \rightsquigarrow (\theta, \eta_1')$. By Lem. 214 we know $\mathit{nextsplit}(\eta_1) = \mathit{nextsplit}(\eta_1; C_2)$, thus $\#\mathit{nextsplit}(\eta_1) = \#\mathit{nextsplit}(\eta_1; C_2) > 1$. From $\eta_1 \rightsquigarrow (\theta, \eta_1')$ we know $\eta_1 \hookrightarrow (\theta, \eta_1')$.

Lemma 216. *For all R, G, I, P, Q, C_2, n, if $\mathbf{Id} \Rightarrow G$ and $(\delta(C_2) \otimes \mu, R, I) \Longmapsto_{\square}^{n} (G, Q)$ for all μ such that $\mu \models I \wedge P$, then for all η, if $(\eta, R, I) \Longmapsto_{\mathrm{ST}}^{n} (G, P)$, then $(\eta; C_2, R, I) \Longmapsto_{\square}^{n} (G, Q)$.*

Proof. For all R, G, I, P, Q, C_2, n such that $\mathbf{Id} \Rightarrow G$ and $(\delta(C_2) \otimes \mu, R, I) \Longmapsto_{\square}^{n} (G, Q)$ for all μ such that $\mu \models I \wedge P$, we prove for all η, if $(\eta, R, I) \Longmapsto_{\mathrm{ST}}^{n} (G, P)$, then $(\eta; C_2, R, I) \Longmapsto_{\square}^{n} (G, Q)$ by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all η, if $(\eta, R, I) \Longmapsto_{\mathrm{ST}}^{k} (G, P)$, then $(\eta; C_2, R, I) \Longmapsto_{\square}^{k} (G, Q)$.
  For all η such that $(\eta, R, I) \Longmapsto_{\mathrm{ST}}^{k+1} (G, P)$, we need to prove
  - (when $\square = \mathrm{ST}$) ${\eta; C_2}^{(\mathit{Stmt})}(\mathbf{skip}) = 1$ or ${\eta; C_2}^{(\mathit{Stmt})}(\mathbf{skip}) = 0$.
    $\eta; C_2^{(\mathit{Stmt})}(\mathbf{skip}) = \sum_\sigma (\eta; C_2)(\mathbf{skip}, \sigma) = 0$.
  - if $\eta; C_2^{(\mathit{Stmt})}(\mathbf{skip}) > 0$, then ${\eta; C_2|_{\mathbf{skip}}}^{(\mathit{State})} \models Q$.
    $\eta; C_2^{(\mathit{Stmt})}(\mathbf{skip}) > 0$ contradicts with $\eta; C_2^{(\mathit{Stmt})} = 0$.

- $\eta; {C_2}^{(State)} \models I$.
  From $(\eta, R, I) \Longmapsto_{\text{ST}}^{k+1} (G, P)$ we know $\eta^{(State)} \models I$. By Lem. 201 we know $\eta; {C_2}^{(State)} = \eta^{(State)} \models I$.

- for all η', if $\eta; C_2 \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longmapsto_{\square}^{k} (G, Q)$.

  For all η' such that $\eta; C_2 \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, by Lem. 209 there exists η'' such that $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta''$ and $\eta' = \eta''; C_2$. From $(\eta, R, I) \Longmapsto_{\text{ST}}^{k+1} (G, P)$ we know $(\eta'', R, I) \Longmapsto_{\text{ST}}^{k} (G, P)$. By IH we have $(\eta''; C_2, R, I) \Longmapsto_{\square}^{k} (G, Q)$. From $\eta' = \eta''; C_2$ we know $(\eta', R, I) \Longmapsto_{\square}^{k} (G, Q)$.

- for all θ and η', if $\eta; C_2 \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, ${\eta'}^{(State)} \models I$ and $(\eta', R, I) \Longmapsto_{\square}^{k} (G, Q)$.
  For all θ and η' such that $\eta; C_2 \hookrightarrow (\theta, \eta')$, from $(\eta, R, I) \Longmapsto_{\text{ST}}^{k+1} (G, P)$ we know $\eta^{(Stmt)}(\mathbf{skip})$
  $= 1$ or $\eta^{(Stmt)}(\mathbf{skip}) = 0$. We prove the two cases respectively.

  * $\eta^{(Stmt)}(\mathbf{skip}) = 1$.
    By Lem. 25 we know $\eta^{(Stmt)} = \delta(\mathbf{skip})$. By Lem. 190 we know $\mathit{nextsplit}(\eta) = \{\mathit{nextsplit}(\mathbf{skip})\} = \{\mathbf{split}(\text{true})\}$. From $\eta; C_2 \hookrightarrow (\theta, \eta')$ by Lem. 191 we know $\eta; C_2 \rightsquigarrow (\theta, \eta')$. From $\eta^{(Stmt)} = \delta(\mathbf{skip})$ by Lem. 210 we know $\eta; C_2 \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta)\}, \delta(C_2) \otimes \eta^{(State)})$. From $\eta; C_2 \rightsquigarrow (\theta, \eta')$ by Lem. 193 we know $\theta = \{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta)\} \subseteq [\![\mathbf{Id}]\!] \subseteq [\![G]\!]$ and $\eta' = \delta(C_2) \otimes \eta^{(State)}$. By Lem. 18 we know ${\eta'}^{(State)} = \eta^{(State)} \models I$. From $(\eta, R, I) \Longmapsto_{\text{ST}}^{k+1} (G, P)$ and $\eta^{(Stmt)}(\mathbf{skip}) = 1 > 0$ we know ${\eta|_{\mathbf{skip}}}^{(State)} \models P$. By Lem. 199 we know $\eta|_{\mathbf{skip}} = \eta$, thus $\eta^{(State)} \models I \wedge P$. From $(\delta(C_2) \otimes \mu, R, I) \Longmapsto_{\square}^{n} (G, Q)$ for all μ such that $\mu \models I \wedge P$ we know $(\delta(C_2) \otimes \eta^{(State)}, R, I) \Longmapsto_{\square}^{n} (G, Q)$, i.e., $(\eta', R, I) \Longmapsto_{\square}^{k+1} (G, Q)$. By Lem. 211 we know $(\eta, R, I) \Longmapsto_{\square}^{k} (G, Q)$.
  * $\eta^{(Stmt)}(\mathbf{skip}) = 0$.
    From $\eta; C_2 \hookrightarrow (\theta, \eta')$ by Lem. 215 there exists η'' such that $\eta' = \eta''; C_2$ and $\eta \hookrightarrow (\theta, \eta'')$. From $(\eta, R, I) \Longmapsto_{\text{ST}}^{k+1} (G, P)$ we know $\theta \subseteq [\![G]\!]$, ${\eta''}^{(State)} \models I$ and $(\eta', R, I) \Longmapsto_{\text{ST}}^{k} (G, P)$. From $\eta' = \eta''; C_2$ and Lem. 201 we know ${\eta'}^{(State)} = \eta''; {C_2}^{(State)} = {\eta''}^{(State)} \models I$. From $(\eta', R, I) \Longmapsto_{\text{ST}}^{k} (G, P)$ by IH we have $(\eta'; C_2, R, I) \Longmapsto_{\square}^{k} (G, Q)$, i.e., $(\eta', R, I) \Longmapsto_{\square}^{k} (G, Q)$.

Lemma 217. *For all* C_1, C_2, μ, $\mathit{init}(C_1; C_2, \mu) = \mathit{init}(C_1, \mu); C_2$.

Proof. For all C_1, C_2, μ,

$$
\begin{aligned}
& \mathit{init}(C_1; C_2, \mu) \\
&= \delta(C_1; C_2) \otimes \mu \\
&= \lambda(C, \sigma).\ \delta(C_1; C_2)(C) \cdot \mu(\sigma) \\
&= \lambda(C, \sigma).\ \begin{cases} \mu(\sigma), & \text{if } C = C_1; C_2 \\ 0, & \text{otherwise} \end{cases} \\
&= \lambda(C, \sigma).\ \begin{cases} \delta(C_1)(C_1') \cdot \mu(\sigma), & \text{if } C = C_1'; C_2 \\ 0, & \text{otherwise} \end{cases} \\
&= (\delta(C_1) \otimes \mu); C_2 \\
&= \mathit{init}(C_1, \mu); C_2.
\end{aligned}
$$

Lemma 218 (Soundness of (SEQ-ST) rule). *For all $C_1, C_2, R, G, I, P, M, Q$, if $R, G, I \models_{ST} \{P\} C_1 \{M\}$, $R, G, I \models_{\square} \{M\} C_2 \{Q\}$ and $\mathbf{Id} \Rightarrow G$, then $R, G, I \models_{\square} \{P\} C_1; C_2 \{Q\}$.*

Proof. For all $C_1, C_2, R, G, I, P, M, Q$ such that $R, G, I \models_{ST} \{P\} C_1 \{M\}$, $R, G, I \models_{\square} \{M\} C_2 \{Q\}$ and $\mathbf{Id} \Rightarrow G$, we need to prove for all μ, if $\mu \models I \wedge P$, then $(\mathit{init}(C_1; C_2, \mu), R, I) \Longrightarrow_{\square}^{n} (G, Q)$ for all n. For all n, from $R, G, I \models_{ST} \{P\} C_1 \{M\}$ and $\mu \models I \wedge P$ we know $(\mathit{init}(C_1, \mu), R, I) \Longrightarrow_{ST}^{n} (G, P)$. From $R, G, I \models_{\square} \{M\} C_2 \{Q\}$ we know $(\delta(C_2) \otimes \mu, R, I) \Longrightarrow_{\square}^{n} (G, Q)$ for all μ such that $\mu \models I \wedge P$. From $(\mathit{init}(C_1, \mu), R, I) \Longrightarrow_{ST}^{n} (G, P)$ and $\mathbf{Id} \Rightarrow G$ by Lem. 216 we know $(\mathit{init}(C_1, \mu); C_2, R, I) \Longrightarrow_{\square}^{n} (G, Q)$. By Lem. 217 we know $\mathit{init}(C_1; C_2, \mu) = \mathit{init}(C_1, \mu); C_2$, thus $(\mathit{init}(C_1; C_2, \mu), R, I) \Longrightarrow_{\square}^{n} (G, Q)$.

Lemma 219. *For all η, b, C, if $\eta^{(Stmt)} = \delta(\textbf{\textit{while}}\ (b)\ \textbf{\textit{do}}\ C)$ and $\eta^{(State)} \models \lceil b \rceil$, then $\eta \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}, \delta(C; \textbf{\textit{while}}\ (b)\ \textbf{\textit{do}}\ C) \otimes \eta^{(State)})$.*

Proof. For all η, b, C such that $\eta^{(Stmt)} = \delta(\textbf{while}\ (b)\ \textbf{do}\ C)$ and $\eta^{(State)} \models \lceil b \rceil$, by Lem. 189 and Lem. 196 we know $C_1 = \textbf{while}\ (b)\ \textbf{do}\ C$ and $\sigma \models b$ for all $(C_1, \sigma) \in \mathit{supp}(\eta)$, thus

$$
\begin{aligned}
& \lambda(C', \sigma').\ \textstyle\sum_{C_1, \sigma} \{\eta(C_1, \sigma) \cdot p \mid (C_1, \sigma) \xrightarrow{p} (C', \sigma')\} \\
&= \lambda(C', \sigma').\ \textstyle\sum_{C_1, \sigma} \{\eta(C_1, \sigma) \cdot p \mid (C_1, \sigma) \in \mathit{supp}(\eta) \wedge (C_1, \sigma) \xrightarrow{p} (C', \sigma')\} \\
&= \lambda(C', \sigma').\ \textstyle\sum_{C_1, \sigma} \{\eta(C_1, \sigma) \cdot p \mid (C_1, \sigma) \in \mathit{supp}(\eta) \wedge C_1 = \textbf{while}\ (b)\ \textbf{do}\ C \wedge \\
&\qquad \sigma \models b \wedge (C_1, \sigma) \xrightarrow{p} (C', \sigma')\} \\
&= \lambda(C', \sigma').\ \textstyle\sum_{\sigma} \{\eta^{(State)}(\sigma) \cdot p \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma \models b \wedge \\
&\qquad (\textbf{while}\ (b)\ \textbf{do}\ C, \sigma) \xrightarrow{p} (C', \sigma')\} \\
&= \lambda(C', \sigma').\ \textstyle\sum_{\sigma} \{\eta^{(State)}(\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma \models b \wedge C' = C; \textbf{while}\ (b)\ \textbf{do}\ C \wedge \sigma' = \sigma\} \\
&= \lambda(C', \sigma').\ \textstyle\sum_{\sigma} \{\eta^{(State)}(\sigma) \mid C' = C; \textbf{while}\ (b)\ \textbf{do}\ C \wedge \sigma' = \sigma\} \\
&= \lambda(C', \sigma').\ \delta(C; \textbf{while}\ (b)\ \textbf{do}\ C)(C') \cdot \eta^{(State)}(\sigma') \\
&= \delta(C; \textbf{while}\ (b)\ \textbf{do}\ C) \otimes \eta^{(State)}
\end{aligned}
$$

and

$$
\begin{aligned}
&\{(\sigma,\sigma') \mid \exists C_1, C'.\ \eta(C_1,\sigma) > 0 \wedge (C_1,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
&= \{(\sigma,\sigma') \mid \exists C_1, C'.\ (C_1,\sigma) \in \mathit{supp}(\eta) \wedge (C_1,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
&= \{(\sigma,\sigma') \mid \exists C_1, C'.\ (C_1,\sigma) \in \mathit{supp}(\eta) \wedge (C_1 = \textbf{while}\ (b)\ \textbf{do}\ C) \wedge\\
&\qquad \sigma \models b \wedge (C_1,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
&= \{(\sigma,\sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma \models b \wedge \exists C'.\ (\textbf{while}\ (b)\ \textbf{do}\ C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
&= \{(\sigma,\sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \exists C'.\ C' = C; \textbf{while}\ (b)\ \textbf{do}\ C \wedge \sigma' = \sigma\}\\
&= \{(\sigma,\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}.
\end{aligned}
$$

Therefore $\eta \rightsquigarrow (\{(\sigma,\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}, \delta(C; \textbf{while}\ (b)\ \textbf{do}\ C) \otimes \eta^{(State)})$.

Lemma 220. *For all η, b, C, if $\eta^{(Stmt)} = \delta(\textbf{\textit{while}}\ (b)\ \textbf{\textit{do}}\ C)$ and $\eta^{(State)} \models \lceil b \rceil$, then $\eta \rightsquigarrow (\{(\sigma,\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}, \delta(\textbf{\textit{skip}}) \otimes \eta^{(State)})$.*

Proof. For all η, b, C such that $\eta^{(Stmt)} = \delta(\textbf{while}\ (b)\ \textbf{do}\ C)$ and $\eta^{(State)} \models \lceil \neg b \rceil$, by Lem. 189 and Lem. 196 we know $C_1 = \textbf{while}\ (b)\ \textbf{do}\ C$ and $\sigma \models \neg b$ for all $(C_1,\sigma) \in \mathit{supp}(\eta)$, thus

$$
\begin{aligned}
&\lambda(C',\sigma').\textstyle\sum_{C_1,\sigma}\{\eta(C_1,\sigma) \cdot p \mid (C_1,\sigma) \xrightarrow{p} (C',\sigma')\}\\
&= \lambda(C',\sigma').\textstyle\sum_{C_1,\sigma}\{\eta(C_1,\sigma) \cdot p \mid (C_1,\sigma) \in \mathit{supp}(\eta) \wedge (C_1,\sigma) \xrightarrow{p} (C',\sigma')\}\\
&= \lambda(C',\sigma').\textstyle\sum_{C_1,\sigma}\{\eta(C_1,\sigma) \cdot p \mid (C_1,\sigma) \in \mathit{supp}(\eta) \wedge C_1 = \textbf{while}\ (b)\ \textbf{do}\ C \wedge\\
&\qquad \sigma \models \neg b \wedge (C_1,\sigma) \xrightarrow{p} (C',\sigma')\}\\
&= \lambda(C',\sigma').\textstyle\sum_{\sigma}\{\eta^{(State)}(\sigma) \cdot p \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma \models \neg b \wedge\\
&\qquad (\textbf{while}\ (b)\ \textbf{do}\ C,\sigma) \xrightarrow{p} (C',\sigma')\}\\
&= \lambda(C',\sigma').\textstyle\sum_{\sigma}\{\eta^{(State)}(\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma \models \neg b \wedge C' = \textbf{skip} \wedge \sigma' = \sigma\}\\
&= \lambda(C',\sigma').\textstyle\sum_{\sigma}\{\eta^{(State)}(\sigma) \mid C' = \textbf{skip} \wedge \sigma' = \sigma\}\\
&= \lambda(C',\sigma').\ \delta(\textbf{skip})(C') \cdot \eta^{(State)}(\sigma')\\
&= \delta(\textbf{skip}) \otimes \eta^{(State)}
\end{aligned}
$$

and

$$
\begin{aligned}
&\{(\sigma,\sigma') \mid \exists C_1, C'.\ \eta(C_1,\sigma) > 0 \wedge (C_1,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
&= \{(\sigma,\sigma') \mid \exists C_1, C'.\ (C_1,\sigma) \in \mathit{supp}(\eta) \wedge (C_1,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
&= \{(\sigma,\sigma') \mid \exists C_1, C'.\ (C_1,\sigma) \in \mathit{supp}(\eta) \wedge (C_1 = \textbf{while}\ (b)\ \textbf{do}\ C) \wedge\\
&\qquad \sigma \models \neg b \wedge (C_1,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
&= \{(\sigma,\sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma \models \neg b \wedge \exists C'.\ (\textbf{while}\ (b)\ \textbf{do}\ C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
&= \{(\sigma,\sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \exists C'.\ C' = \textbf{skip} \wedge \sigma' = \sigma\}\\
&= \{(\sigma,\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}.
\end{aligned}
$$

Therefore $\eta \rightsquigarrow (\{(\sigma,\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}, \delta(\textbf{skip}) \otimes \eta^{(State)})$.

Lemma 221 (Soundness of (WHILE-ST) rule). *For all $b, C, R, G, I, P_1, P_2, Q$, if $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $\mathbf{Sta}(Q, R, I)$, $P_1 \Rightarrow \lceil b \rceil$, $P_2 \Rightarrow \lceil \neg b \rceil \wedge Q$, $\mathbf{Id} \Rightarrow G$ and $R, G, I \models_{ST} \{P_1\} C \{P_1 \vee P_2\}$, then $R, G, I \models_{ST} \{P_1 \vee P_2\} \textbf{\textit{while}}\ (b)\ \textbf{\textit{do}}\ C \{Q\}$.*

Proof. For all $b, C, R, G, I, P_1, P_2, Q$ such that $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $\mathbf{Sta}(Q, R, I)$, $P_1 \Rightarrow \lceil b \rceil$, $P_2 \Rightarrow \lceil \neg b \rceil \wedge Q$, $\mathbf{Id} \Rightarrow G$ and $R, G, I \models_{\text{ST}} \{P_1\} C \{P_1 \vee P_2\}$, we need to prove for all μ, if $\mu \models I \wedge P$, then $(\mathit{init}(\mathbf{while}\ (b)\ \mathbf{do}\ C, \mu), R, I) \Longrightarrow^n_{\text{ST}} (G, Q)$ for all n. For all μ such that $\mu \models I \wedge (P_1 \vee P_2)$, by Lem. 18 we know $\mathit{init}(\mathbf{while}\ (b)\ \mathbf{do}\ C, \mu)^{(Stmt)} = (\delta(\mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes \mu)^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$. To prove $(\mathit{init}(\mathbf{while}\ (b)\ \mathbf{do}\ C, \mu), R, I) \Longrightarrow^n_{\text{ST}} (G, Q)$ for all n, it suffices to prove for all n and η, if $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ and $\eta^{(State)} \models I \wedge (P_1 \vee P_2)$, then $(\eta, R, I) \Longrightarrow^n_{\text{ST}} (G, Q)$. We prove by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all η, if $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ and $\eta^{(State)} \models I \wedge (P_1 \vee P_2)$, then $(\eta, R, I) \Longrightarrow^k_{\text{ST}} (G, Q)$.
  For all η such that $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ and $\eta^{(State)} \models I \wedge (P_1 \vee P_2)$, to prove $(\eta, R, I) \Longrightarrow^{k+1}_{\text{ST}} (G, Q)$, we need to prove
  - $\eta^{(Stmt)}(\mathbf{skip}) = 0$ or $\eta^{(Stmt)}(\mathbf{skip}) = 1$.
    From $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ we have $\eta^{(Stmt)}(\mathbf{skip}) = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)(\mathbf{skip}) = 0$.
  - if $\eta^{(Stmt)}(\mathbf{skip}) > 0$, then $\eta|_{\mathbf{skip}}{}^{(State)} \models Q$.
    $\eta^{(Stmt)}(\mathbf{skip}) > 0$ contradicts with $\eta^{(Stmt)}(\mathbf{skip}) = 0$.
  - $\eta^{(State)} \models I$.
    From $\eta^{(State)} \models I \wedge (P_1 \vee P_2)$ we have $\eta^{(State)} \models I$.
  - for all η', if $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longrightarrow^k_{\text{ST}} (G, Q)$.
    For all η' such that $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, by Lem. 188 we know $\mathit{supp}(\eta'^{(Stmt)}) \subseteq \mathit{supp}(\eta^{(Stmt)})$. From $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ by Lem. 27 we know $\eta'^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$. From $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $\eta^{(State)} \models I \wedge (P_1 \vee P_2)$ and $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$ by Lem. 186 we have $\eta'^{(State)} \models I \wedge (P_1 \vee P_2)$. From $\eta'^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ by IH we have $(\eta', R, I) \Longrightarrow^k_{\text{ST}} (G, Q)$.
  - for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, $\eta'^{(State)} \models I$ and $(\eta', R, I) \Longrightarrow^k_{\text{ST}} (G, Q)$.
    For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, from $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ by Lem. 190 we have $\mathit{nextsplit}(\eta) = \{\mathit{nextsplit}(\mathbf{while}\ (b)\ \mathbf{do}\ C)\} = \{\mathbf{split}(\text{true})\}$. From $\eta \hookrightarrow (\theta, \eta')$ by Lem. 191 we know $\eta \rightsquigarrow (\theta, \eta')$. From $\eta^{(State)} \models I \wedge (P_1 \vee P_2)$ we know $\eta^{(State)} \models I \wedge P_1$ or $\eta^{(State)} \models I \wedge P_2$. We prove the two cases respectively.
    - case 1: $\eta^{(State)} \models I \wedge P_1$.
      From $P_1 \Rightarrow \lceil b \rceil$ we know $\eta^{(State)} \models \lceil b \rceil$. By Lem. 219 we know $\eta \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}, \delta(C; \mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes \eta^{(State)})$. From $\eta \rightsquigarrow (\theta, \eta')$ by Lem. 193 we have $\theta = \{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}$ and $\eta' = \delta(C; \mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes \eta^{(State)}$, thus $\theta \subseteq [\![\mathbf{Id}]\!] \subseteq [\![G]\!]$ and $\eta'^{(State)} = \eta^{(State)} \models I$. From $R, G, I \models_{\text{ST}} \{P_1\} C \{P_1 \vee P_2\}$ we know $\eta^{(State)} \models I \wedge P_1$ we have $(\mathit{init}(C, \mu), R, I) \Longrightarrow^k_{\text{ST}} (G, P_1 \vee P_2)$. From IH we know $(\delta(\mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes \mu, R, I) \Longrightarrow^k_{\text{ST}} (G, Q)$ for all μ such

that $\mu \models I \wedge (P_1 \vee P_2)$. From $(\mathit{init}(C,\mu), R, I) \Longrightarrow^{n}_{\mathrm{ST}} (G, P_1 \vee P_2)$ and $\mathbf{Id} \Rightarrow G$ by Lem. 216 we know $(\mathit{init}(C,\mu); \mathbf{while}\ (b)\ \mathbf{do}\ C, R, I) \Longrightarrow^{n}_{\mathrm{ST}} (G, Q)$. By Lem. 217 we know $\eta' = \delta(C; \mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes \eta^{(State)} = \mathit{init}(C; \mathbf{while}\ (b)\ \mathbf{do}\ C, \eta^{(State)}) = \mathit{init}(C, \eta^{(State)}); \mathbf{while}\ (b)\ \mathbf{do}\ C$, thus $(\eta', R, I) \Longrightarrow^{k}_{\mathrm{ST}} (G, Q)$.

* case 2: $\eta^{(State)} \models I \wedge P_2$.
  From $P_2 \Rightarrow \lceil \neg b \rceil \wedge Q$ we know $\eta^{(State)} \models \lceil \neg b \rceil \wedge Q$. By Lem. 220 we know $\eta \rightsquigarrow (\{(\sigma,\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}, \delta(\mathbf{skip}) \otimes \eta^{(State)})$. From $\eta \rightsquigarrow (\theta, \eta')$ by Lem. 193 we have $\theta = \{(\sigma,\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\}$ and $\eta' = \delta(\mathbf{skip}) \otimes \eta^{(State)}$, thus $\theta \subseteq [\![\mathbf{Id}]\!] \subseteq [\![G]\!]$, ${\eta'}^{(Stmt)} = \delta(\mathbf{skip})$ and ${\eta'}^{(State)} = \eta^{(State)} \models I \wedge Q$. From $\mathbf{Sta}(Q, R, I)$, $\mathbf{Id} \Rightarrow G$, ${\eta'}^{(Stmt)} = \delta(\mathbf{skip})$ and ${\eta'}^{(State)} \models I \wedge Q$ by Lem. 194 we know $(\eta', R, I) \Longrightarrow^{k}_{\mathrm{ST}} (G, Q)$.

Lemma 222. *For all η, b, C, if $\eta^{(Stmt)} = \delta(\langle C \rangle)$, then $\eta \rightsquigarrow (\{(\sigma,\sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma' \in \mathit{supp}([\![C]\!](\sigma))\}, \delta(\boldsymbol{skip}) \otimes [\![C]\!](\eta^{(State)}))$.*

Proof. For all η, b, C such that $\eta^{(Stmt)} = \delta(\langle C \rangle)$ and $\eta^{(State)} \models \lceil \neg b \rceil$, by Lem. 189 and Lem. 196 we know $C_1 = \langle C \rangle$ for all $(C_1, \sigma) \in \mathit{supp}(\eta)$, thus

$$
\begin{aligned}
&\lambda(C',\sigma').\textstyle\sum_{C_1,\sigma}\{\eta(C_1,\sigma)\cdot p \mid (C_1,\sigma) \xrightarrow{p} (C',\sigma')\}\\
=\ &\lambda(C',\sigma').\textstyle\sum_{C_1,\sigma}\{\eta(C_1,\sigma)\cdot p \mid (C_1,\sigma) \in \mathit{supp}(\eta) \wedge (C_1,\sigma) \xrightarrow{p} (C',\sigma')\}\\
=\ &\lambda(C',\sigma').\textstyle\sum_{C_1,\sigma}\{\eta(C_1,\sigma)\cdot p \mid (C_1,\sigma) \in \mathit{supp}(\eta) \wedge C_1 = \langle C\rangle \wedge (C_1,\sigma) \xrightarrow{p} (C',\sigma')\}\\
=\ &\lambda(C',\sigma').\textstyle\sum_{\sigma}\{\eta^{(State)}(\sigma)\cdot p \mid (\langle C\rangle,\sigma) \xrightarrow{p} (C',\sigma')\}\\
=\ &\lambda(C',\sigma').\ \delta(\mathbf{skip})(C')\cdot\textstyle\sum_{\sigma}\{\eta^{(State)}(\sigma)\cdot p \mid \exists k.\forall n \geq k.\ (C,\sigma) \xrightarrow{p}{}^{n} (\mathbf{skip},\sigma')\}\\
=\ &\lambda(C',\sigma').\ \delta(\mathbf{skip})(C')\cdot\textstyle\sum_{\sigma}\eta^{(State)}(\sigma)\cdot[\![C]\!](\sigma)(\sigma')\\
=\ &\lambda(C',\sigma').\ \delta(\mathbf{skip})(C')\cdot\mathbb{E}_{\sigma\sim\eta^{(State)}}\{[\![C]\!](\sigma)\}(\sigma')\\
=\ &\lambda(C',\sigma').\ \delta(\mathbf{skip})(C')\cdot[\![C]\!](\eta^{(State)})(\sigma')\\
=\ &\delta(\mathbf{skip})\otimes[\![C]\!](\eta^{(State)})
\end{aligned}
$$

and

$$
\begin{aligned}
&\{(\sigma,\sigma') \mid \exists C_1, C'.\ \eta(C_1,\sigma) > 0 \wedge (C_1,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
=\ &\{(\sigma,\sigma') \mid \exists C_1, C'.\ (C_1,\sigma) \in \mathit{supp}(\eta) \wedge (C_1,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
=\ &\{(\sigma,\sigma') \mid \exists C_1, C'.\ (C_1,\sigma) \in \mathit{supp}(\eta) \wedge (C_1 = \langle C\rangle) \wedge (C_1,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
=\ &\{(\sigma,\sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \exists C'.\ (\langle C\rangle,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}\\
=\ &\{(\sigma,\sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \exists k.\forall n \geq k.\ (C,\sigma) \xrightarrow{p}{}^{n} (\mathbf{skip},\sigma') \wedge p > 0\}\\
=\ &\{(\sigma,\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge [\![C]\!](\sigma)(\sigma') > 0\}\\
=\ &\{(\sigma,\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma' \in \mathit{supp}([\![C]\!](\sigma))\}.
\end{aligned}
$$

Therefore $\eta \rightsquigarrow (\{(\sigma,\sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma' \in \mathit{supp}([\![C]\!](\sigma))\}, \delta(\mathbf{skip}) \otimes [\![C]\!](\eta^{(State)}))$.

Lemma 223 (Soundness of (ATOM) rule). *For all C, R, G, I, P, Q, if $\mathbf{Sta}(P, R, I)$, $\mathbf{Sta}(Q, R, I)$, $G \models_{sQ} \{I \wedge P\} C \{I \wedge Q\}$ and $\mathbf{Id} \Rightarrow G$, then $R, G, I \models_{ST} \{P\}\langle C\rangle\{Q\}$.*

Proof. For all C, R, G, I, P, Q such that $\mathbf{Sta}(P, R, I)$, $\mathbf{Sta}(Q, R, I)$, $G \models_{\text{SQ}} \{I \wedge P\}C\{I \wedge Q\}$ and $\mathbf{Id} \Rightarrow G$, to prove $R, G, I \models_{\text{ST}} \{P\}\langle C\rangle\{Q\}$, we need to prove for all μ, if $\mu \models I \wedge P$, then $(\mathit{init}(\langle C\rangle, \mu), R, I) \Longrightarrow^{n}_{\text{ST}} (G, Q)$ for all n. For all μ such that $\mu \models I \wedge P$, by Lem. 18 we know $\mathit{init}(\langle C\rangle, \mu)^{(\mathit{Stmt})} = (\delta(\langle C\rangle) \otimes \mu)^{(\mathit{Stmt})} = \delta(\langle C\rangle)$. To prove $(\mathit{init}(\langle C\rangle, \mu), R, I) \Longrightarrow^{n}_{\text{ST}} (G, Q)$ for all n, it suffices to prove for all n and η, if $\eta^{(\mathit{Stmt})} = \delta(\langle C\rangle)$ and $\eta^{(\mathit{State})} \models I \wedge P$, then $(\eta, R, I) \Longrightarrow^{n}_{\text{ST}} (G, Q)$. We prove by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all η, if $\eta^{(\mathit{Stmt})} = \delta(\langle C\rangle)$ and $\eta^{(\mathit{State})} \models I \wedge P$, then $(\eta, R, I) \Longrightarrow^{k}_{\text{ST}} (G, Q)$.
  For all η such that $\eta^{(\mathit{Stmt})} = \delta(\langle C\rangle)$ and $\eta^{(\mathit{State})} \models I \wedge P$, we need to prove
  - $\eta^{(\mathit{Stmt})}(\mathbf{skip}) = 1$ or $\eta^{(\mathit{Stmt})}(\mathbf{skip}) = 0$.
    From $\eta^{(\mathit{Stmt})} = \delta(\langle C\rangle)$ we know $\eta^{(\mathit{Stmt})}(\mathbf{skip}) = \delta(\langle C\rangle)(\mathbf{skip}) = 0$.
  - if $\eta^{(\mathit{Stmt})}(\mathbf{skip}) > 0$, then $\eta|_{\mathbf{skip}}{}^{(\mathit{State})} \models Q$.
    $\eta^{(\mathit{Stmt})}(\mathbf{skip}) > 0$ contradicts with $\eta^{(\mathit{Stmt})}(\mathbf{skip}) = 0$.
  - $\eta^{(\mathit{State})} \models I$.
    From $\eta^{(\mathit{State})} \models I \wedge P$ we know $\eta^{(\mathit{State})} \models I$.
  - for all η', if $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longrightarrow^{k}_{\square} (G, Q)$.
    For all η' such that $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, by Lem. 188 we know $\mathit{supp}(\eta'^{(\mathit{Stmt})}) \subseteq \mathit{supp}(\eta^{(\mathit{Stmt})})$. From $\eta^{(\mathit{Stmt})} = \delta(\langle C\rangle)$ by Lem. 27 we know $\eta'^{(\mathit{Stmt})} = \delta(\langle C\rangle)$. From $\mathbf{Sta}(P, R, I)$, $\eta^{(\mathit{State})} \models I \wedge P$ and $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$ by Lem. 186 we have $\eta'^{(\mathit{State})} \models I \wedge P$. From $\eta'^{(\mathit{Stmt})} = \delta(\langle C\rangle)$ by IH we have $(\eta', R, I) \Longrightarrow^{k}_{\text{ST}} (G, Q)$.
  - for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, $\eta'^{(\mathit{State})} \models I$ and $(\eta', R, I) \Longrightarrow^{k}_{\square} (G, Q)$.
    For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, from $\eta^{(\mathit{Stmt})} = \delta(\langle C\rangle)$ by Lem. 190 we have $\mathit{nextsplit}(\eta) = \{\mathit{nextsplit}(\langle C\rangle)\} = \{\mathbf{split}(\text{true})\}$. From $\eta \hookrightarrow (\theta, \eta')$ by Lem. 191 we know $\eta \rightsquigarrow (\theta, \eta')$. From $\eta^{(\mathit{Stmt})} = \delta(\langle C\rangle)$ by Lem. 222 we know $\eta \rightsquigarrow (\{(\sigma, \sigma') \mid \sigma \in \mathit{supp}(\eta^{(\mathit{State})}) \wedge \sigma' \in \mathit{supp}([\![C]\!](\sigma))\}, \delta(\mathbf{skip}) \otimes [\![C]\!](\eta^{(\mathit{State})}))$. From $\eta \rightsquigarrow (\theta, \eta')$ by Lem. 193 we know $\theta = \{(\sigma, \sigma') \mid \sigma \in \mathit{supp}(\eta^{(\mathit{State})}) \wedge \sigma' \in \mathit{supp}([\![C]\!](\sigma))\}$ and $\eta' = \delta(\mathbf{skip}) \otimes [\![C]\!](\eta^{(\mathit{State})})$. By Lem. 18 and Lem. 19 we know $\eta'^{(\mathit{Stmt})} = \delta(\mathbf{skip})$ and $\eta'^{(\mathit{State})} = [\![C]\!](\eta^{(\mathit{State})})$. From $|\eta'^{(\mathit{State})}| = 1$ we know $|[\![C]\!](\eta^{(\mathit{State})})| = 1$. From $\models_{\text{SQ}} \{I \wedge P\}C\{I \wedge Q\}$ and $\eta^{(\mathit{State})} \models I \wedge P$ we know $[\![C]\!](\eta^{(\mathit{State})}) \models I \wedge Q$ and $(\sigma, \sigma') \models G$ for all $\sigma \in \mathit{supp}(\eta^{(\mathit{State})})$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$, thus $\theta = \{(\sigma, \sigma') \mid \sigma \in \mathit{supp}(\eta^{(\mathit{State})}) \wedge \sigma' \in \mathit{supp}([\![C]\!](\sigma))\} \subseteq [\![G]\!]$ and $\eta'^{(\mathit{State})} = [\![C]\!](\eta^{(\mathit{State})}) \models I \wedge Q$. From $\mathbf{Sta}(Q, R, I)$, $\mathbf{Id} \Rightarrow G$, $\eta'^{(\mathit{Stmt})} = \delta(\mathbf{skip})$ and $\eta'^{(\mathit{State})} \models I \wedge Q$ by Lem. 194 we know $(\eta', R, I) \Longrightarrow^{k}_{\text{ST}} (G, Q)$.

Lemma 224. *For all η, b, C, if $\eta^{(\mathit{Stmt})} = \delta(\langle C\rangle\ \boldsymbol{split}(b_1, \ldots, b_k))$, then $\eta \rightsquigarrow (\{(\sigma, \sigma') \mid \sigma \in \mathit{supp}(\eta^{(\mathit{State})}) \wedge \sigma' \in \mathit{supp}([\![C]\!](\sigma))\}, \delta(\boldsymbol{skip}) \otimes [\![C]\!](\eta^{(\mathit{State})}))$.*

Proof. For all η, b, C such that $\eta^{(Stmt)} = \delta(\langle C \rangle)$ and $\eta^{(State)} \models \lceil \neg b \rceil$, by Lem. 189 and Lem. 196 we know $C_1 = \langle C \rangle$ for all $(C_1, \sigma) \in \mathit{supp}(\eta)$, thus

$$
\begin{aligned}
&\lambda(C', \sigma'). \textstyle\sum_{C_1,\sigma} \{\eta(C_1, \sigma) \cdot p \mid (C_1, \sigma) \xrightarrow{p} (C', \sigma')\} \\
&= \lambda(C', \sigma'). \textstyle\sum_{C_1,\sigma} \{\eta(C_1, \sigma) \cdot p \mid (C_1, \sigma) \in \mathit{supp}(\eta) \wedge (C_1, \sigma) \xrightarrow{p} (C', \sigma')\} \\
&= \lambda(C', \sigma'). \textstyle\sum_{C_1,\sigma} \{\eta(C_1, \sigma) \cdot p \mid (C_1, \sigma) \in \mathit{supp}(\eta) \wedge \\
&\qquad C_1 = \langle C \rangle\ \mathbf{split}(b_1, \ldots, b_k) \wedge (C_1, \sigma) \xrightarrow{p} (C', \sigma')\} \\
&= \lambda(C', \sigma'). \textstyle\sum_{\sigma} \{\eta^{(State)}(\sigma) \cdot p \mid (\langle C \rangle\ \mathbf{split}(b_1, \ldots, b_k), \sigma) \xrightarrow{p} (C', \sigma')\} \\
&= \lambda(C', \sigma').\ \delta(\mathbf{skip})(C') \cdot \textstyle\sum_{\sigma} \{\eta^{(State)}(\sigma) \cdot p \mid (\langle C \rangle, \sigma) \xrightarrow{p} (\mathbf{skip}, \sigma')\} \\
&= \lambda(C', \sigma').\ \delta(\mathbf{skip})(C') \cdot \textstyle\sum_{\sigma} \{\eta^{(State)}(\sigma) \cdot p \mid \exists k. \forall n \geq k.\ (C, \sigma) \xrightarrow{p}{}^{n} (\mathbf{skip}, \sigma')\} \\
&= \lambda(C', \sigma').\ \delta(\mathbf{skip})(C') \cdot \textstyle\sum_{\sigma} \eta^{(State)}(\sigma) \cdot [\![C]\!](\sigma)(\sigma') \\
&= \lambda(C', \sigma').\ \delta(\mathbf{skip})(C') \cdot \mathbb{E}_{\sigma \sim \eta^{(State)}} \{[\![C]\!](\sigma)\}(\sigma') \\
&= \lambda(C', \sigma').\ \delta(\mathbf{skip})(C') \cdot [\![C]\!](\eta^{(State)})(\sigma') \\
&= \delta(\mathbf{skip}) \otimes [\![C]\!](\eta^{(State)})
\end{aligned}
$$

and

$$
\begin{aligned}
&\{(\sigma, \sigma') \mid \exists C_1, C'.\ \eta(C_1, \sigma) > 0 \wedge (C_1, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\
&= \{(\sigma, \sigma') \mid \exists C_1, C'.\ (C_1, \sigma) \in \mathit{supp}(\eta) \wedge (C_1, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\
&= \{(\sigma, \sigma') \mid \exists C_1, C'.\ (C_1, \sigma) \in \mathit{supp}(\eta) \wedge (C_1 = \langle C \rangle\ \mathbf{split}(b_1, \ldots, b_k)) \wedge \\
&\qquad (C_1, \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\
&= \{(\sigma, \sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \exists C'.\ (\langle C \rangle\ \mathbf{split}(b_1, \ldots, b_k), \sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\} \\
&= \{(\sigma, \sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge (\langle C \rangle, \sigma) \xrightarrow{p} (\mathbf{skip}, \sigma') \wedge p > 0\} \\
&= \{(\sigma, \sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \exists k. \forall n \geq k.\ (C, \sigma) \xrightarrow{p}{}^{n} (\mathbf{skip}, \sigma') \wedge p > 0\} \\
&= \{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge [\![C]\!](\sigma)(\sigma') > 0\} \\
&= \{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma' \in \mathit{supp}([\![C]\!](\sigma))\}.
\end{aligned}
$$

Therefore $\eta \rightsquigarrow (\{(\sigma, \sigma') \mid \sigma \in \mathit{supp}(\eta^{(State)}) \wedge \sigma' \in \mathit{supp}([\![C]\!](\sigma))\}, \delta(\mathbf{skip}) \otimes [\![C]\!](\eta^{(State)}))$.

Lemma 225. *For all $\mu, Q_1, \ldots, Q_n$, if $\mu \models Q_1 \oplus \cdots \oplus Q_n$, then there exists $\mu_1, \ldots, \mu_n, p_1, \ldots, p_n$ such that $\mu = \lambda\sigma.\ p_1 \cdot \mu_1(\sigma) + \cdots + p_n \cdot \mu_n(\sigma)$ and $\mu_i \models Q_i$ for all i such that $p_i > 0$.*

Proof. by induction on k.

- base case: $n = 1$.
  For all μ and Q_1 such that $\mu \models Q_1$, let $\mu_1 \stackrel{\text{def}}{=} \mu$ and $p_1 = 1$, we have $\mu = \lambda\sigma.\ p_1 \cdot \mu_1(\sigma)$ and $\mu_1 \models Q_1$.
- inductive case: $n = k + 1$.
  IH: for all $\mu, Q_1, \ldots, Q_k$, if $\mu \models Q_1 \oplus \cdots \oplus Q_k$, then there exists $\mu_1, \ldots, \mu_k, p_1, \ldots, p_k$ such that $\mu = \lambda\sigma.\ p_1 \cdot \mu_1(\sigma) + \cdots + p_k \cdot \mu_k(\sigma)$ and $\mu_i \models Q_i$ for all i such that $p_i > 0$. For all $\mu, Q_1, \ldots, Q_{k+1}$ such that $\mu \models Q_1 \oplus \cdots \oplus Q_{k+1}$, there exists p such that $\mu \models (Q_1 \oplus \cdots \oplus Q_k) \oplus_p Q_{k+1}$. There are three cases.
  - $p = 1$ and $\mu \models Q_1 \oplus \cdots \oplus Q_k$.
    From $\mu \models Q_1 \oplus \cdots \oplus Q_k$ by IH there exists $\mu_1, \ldots, \mu_k, p_1, \ldots, p_k$ such

that $\mu = \lambda\sigma.\ p_1 \cdot \mu_1(\sigma) + \cdots + p_k \cdot \mu_k(\sigma)$ and $\mu_i \models Q_i$ for all i such that $p_i > 0$. Let μ_{k+1} be any state distribution and $p_{k+1} \stackrel{\text{def}}{=} 0$, we have $\mu = \lambda\sigma.\ p_1 \cdot \mu_1(\sigma) + \cdots + p_{k+1} \cdot \mu_{k+1}(\sigma)$ and $\mu_i \models Q_i$ for all i such that $p_i > 0$.

- $p = 0$ and $\mu \models Q_{k+1}$.
  Let $p_{k+1} = 1$, $\mu_{k+1} = \mu$, $p_1 = \cdots = p_k = 0$ and $\mu_1, \ldots, \mu_k$ be any state distributions, we have $\mu = \lambda\sigma.\ p_1 \cdot \mu_1(\sigma) + \cdots + p_{k+1} \cdot \mu_{k+1}(\sigma)$ and $\mu_i \models Q_i$ for all i such that $p_i > 0$.
- $0 < p < 1$ and there exists μ' and μ'' μ such that $\mu = \mu' \oplus_p \mu''$, $\mu' \models Q_1 \oplus \cdots \oplus Q_k$ and $\mu'' \models Q_{k+1}$.
  From $\mu' \models Q_1 \oplus \cdots \oplus Q_k$ by IH there exists $\mu_1, \ldots, \mu_k, p'_1, \ldots, p'_k$ such that $\mu = \lambda\sigma.\ p'_1 \cdot \mu_1(\sigma) + \cdots + p'_k \cdot \mu_k(\sigma)$ and $\mu_i \models Q_i$ for all i such that $p'_i > 0$. Let $p_1 \stackrel{\text{def}}{=} p \cdot p'_1, \ldots, p_k \stackrel{\text{def}}{=} p \cdot p'_k$, $p_{k+1} = 1 - p$ and $\mu_{k+1} \stackrel{\text{def}}{=} \mu''$, then $\mu = \mu' \oplus_p \mu'' = \lambda\sigma.\ p \cdot \mu'(\sigma) + (1-p) \cdot \mu''(\sigma) = \lambda\sigma.\ p \cdot (p'_1 \cdot \mu_1(\sigma) + \cdots + p'_k \cdot \mu_k(\sigma)) + (1-p) \cdot \mu''(\sigma) = \lambda\sigma.\ p_1 \cdot \mu_1(\sigma) + \cdots + p_k \cdot \mu_k(\sigma) + p_{k+1} \cdot \mu_{k+1}(\sigma)$. For all i such that $p_i > 0$, we have $i \leq k \wedge p'_i > 0$ or $i = k+1$. If $i \leq k \wedge p'_i > 0$, we know $\mu_i \models Q_i$. Otherwise $i = k + 1$, we know $\mu_i = \mu_{k+1} = \mu''$ and $Q_i = Q_{k+1}$, from $\mu'' \models Q_{k+1}$ we have $\mu_i \models Q_i$.

Lemma 226. *For all μ and b, $\mu \models \lceil b \rceil$ if and only if $[\![\boldsymbol{Pr}(b)]\!]_\mu = 1$.*

Proof. For all μ and b, we have

$$
\begin{aligned}
& [\![\mathbf{Pr}(b)]\!]_\mu = 1 \\
\iff & \mathbf{Pr}_{\sigma\sim\mu}[\sigma \models b] = |\mu| \\
\iff & \textstyle\sum_\sigma \{\mu(\sigma) \mid \sigma \models b\} = \sum_\sigma \mu(\sigma) \\
\iff & \textstyle\sum_\sigma \{\mu(\sigma) \mid \sigma \in \mathit{supp}(\mu) \wedge \sigma \models b\} = \sum_\sigma \{\mu(\sigma) \mid \sigma \in \mathit{supp}(\mu)\} \\
\iff & \textstyle\sum_\sigma \{\mu(\sigma) \mid \sigma \in \mathit{supp}(\mu) \wedge \sigma \models b\} = \sum_\sigma \{\mu(\sigma) \mid \sigma \in \mathit{supp}(\mu) \wedge \sigma \models b\} + \\
& \qquad \textstyle\sum_\sigma \{\mu(\sigma) \mid \sigma \in \mathit{supp}(\mu) \wedge \sigma \not\models b\} \\
\iff & \textstyle\sum_\sigma \{\mu(\sigma) \mid \sigma \in \mathit{supp}(\mu) \wedge \sigma \not\models b\} = 0 \\
\iff & \{\sigma \mid \sigma \in \mathit{supp}(\mu) \wedge \sigma \not\models b\} = \emptyset \\
\iff & \nexists \sigma \in \mathit{supp}(\mu).\ \sigma \not\models b \\
\iff & \forall \sigma \in \mathit{supp}(\mu).\ \sigma \models b \\
\iff & \mu \models \lceil b \rceil.
\end{aligned}
$$

Lemma 227. *For all μ and b, $\mu \models \lceil \neg b \rceil$ if and only if $[\![\boldsymbol{Pr}(b)]\!]_\mu = 0$.*

Proof. For all μ and b, we have

$$
\begin{aligned}
& [\![\mathbf{Pr}(b)]\!]_\mu = 0 \\
\iff & \mathbf{Pr}_{\sigma\sim\mu}[\sigma \models b] = |\mu| \\
\iff & \textstyle\sum_\sigma \{\mu(\sigma) \mid \sigma \models b\} = 0 \\
\iff & \textstyle\sum_\sigma \{\mu(\sigma) \mid \sigma \in \mathit{supp}(\mu) \wedge \sigma \models b\} = 0 \\
\iff & \{\sigma \mid \sigma \in \mathit{supp}(\mu) \wedge \sigma \models b\} = \emptyset \\
\iff & \nexists \sigma \in \mathit{supp}(\mu).\ \sigma \models b \\
\iff & \forall \sigma \in \mathit{supp}(\mu).\ \sigma \models \neg b \\
\iff & \mu \models \lceil \neg b \rceil.
\end{aligned}
$$

Lemma 228. *For all μ and b, if $\mu \models \lceil b \rceil$, then $\mu(\sigma) = 0$ for all σ such that $\sigma \not\models b$.*

Proof. For all μ and b such that $\mu \models \lceil b \rceil$, by Lem. 226 we know $[\![\mathbf{Pr}(b)]\!]_\mu = 1$. From $[\![\mathbf{Pr}(b)]\!]_\mu = \mathbf{Pr}_{\sigma\sim\mu}[\sigma \models b] = \sum_\sigma\{\mu(\sigma) \mid \sigma \models b\}$ and $1 = |\mu| = \sum_\sigma \mu(\sigma) = \sum_\sigma\{\mu(\sigma) \mid \sigma \models b\} + \sum_\sigma\{\mu(\sigma) \mid \sigma \not\models b\}$ we know $\sum_\sigma\{\mu(\sigma) \mid \sigma \models b\} = \sum_\sigma\{\mu(\sigma) \mid \sigma \models b\} + \sum_\sigma\{\mu(\sigma) \mid \sigma \not\models b\}$, thus $\sum_\sigma\{\mu(\sigma) \mid \sigma \not\models b\} = 0$. Therefore $\mu(\sigma) = 0$ for all σ such that $\sigma \not\models b$.

Lemma 229. *For all $\mu, \mu_1, \ldots, \mu_k, p_1, \ldots, p_k, b_1, \ldots, b_k$, if $\mu = \lambda\sigma.\ p_1 \cdot \mu_1(\sigma) + \cdots + p_k \cdot \mu_k(\sigma)$, $\mu_i \models \lceil b_i \rceil$ for all i such that $p_i > 0$, and $\sigma \models \neg(b_i \wedge b_j)$ for all σ, i, j such that $i \neq j$, then $p_j = 0$ or $\mu_j(\sigma) = 0$ for all i, j, σ such that $i \neq j$ and $\sigma \models b_i$.*

Proof. For all $\mu, \mu_1, \ldots, \mu_k, p_1, \ldots, p_k, b_1, \ldots, b_k$ such that $\mu = \lambda\sigma.\ p_1 \cdot \mu_1(\sigma) + \cdots + p_k \cdot \mu_k(\sigma)$, $\mu_i \models \lceil b_i \rceil$ for all i such that $p_i > 0$, and $\sigma \models \neg(b_i \wedge b_j)$ for all σ, i, j such that $i \neq j$. For all i, j, σ such that $i \neq j$ and $\sigma \models b_i$, from $i \neq j$ we have $\sigma \models \neg(b_i \wedge b_j)$, i.e., $\neg(\sigma \models b_i \wedge \sigma \models b_j)$. From $\sigma \models b_i$ we know $\sigma \not\models b_j$. It is obvious that $p_j = 0$ or $p_j > 0$. To prove $p_j = 0$ or $\mu_j(\sigma) = 0$, we need to prove if $p_j > 0$ then $\mu_j(\sigma) = 0$. From $p_j > 0$ we know $\mu_j \models \lceil b_j \rceil$. From $\sigma \not\models b_j$ by Lem. 228 we have $\mu_j(\sigma) = 0$.

Lemma 230. *For all $\mu, \mu_1, \ldots, \mu_k, p_1, \ldots, p_k, b_1, \ldots, b_k$, if $\mu = \lambda\sigma.\ p_1 \cdot \mu_1(\sigma) + \cdots + p_k \cdot \mu_k(\sigma)$, $\mu_i \models \lceil b_i \rceil$ for all i such that $p_i > 0$, and $\sigma \models \neg(b_i \wedge b_j)$ for all σ, i, j such that $i \neq j$, then $p_j = 0$ or $[\![\boldsymbol{Pr}(b_i)]\!]_{\mu_j} = 0$ for all i, j such that $i \neq j$.*

Proof. For all $\mu, \mu_1, \ldots, \mu_k, p_1, \ldots, p_k, b_1, \ldots, b_k$ such that $\mu = \lambda\sigma.\ p_1 \cdot \mu_1(\sigma) + \cdots + p_k \cdot \mu_k(\sigma)$, $\mu_i \models \lceil b_i \rceil$ for all i such that $p_i > 0$, and $\sigma \models \neg(b_i \wedge b_j)$ for all σ, i, j such that $i \neq j$. For all i and j such that $i \neq j$, by Lem. 229 we know $p_j = 0$ or $\mu_j(\sigma) = 0$ for all σ such that $\sigma \models b_i$, thus $p_j = 0$ or $[\![\mathbf{Pr}(b_i)]\!]_{\mu_j} = \mathbf{Pr}_{\sigma\sim\mu_j}[\sigma \models b_i] = \sum_\sigma\{\mu_j(\sigma) \mid \sigma \models b_i\} = 0$.

Lemma 231. *For all $\mu, \mu_1, \ldots, \mu_k, p_1, \ldots, p_k, b_1, \ldots, b_k$, if $\mu = \lambda\sigma.\ p_1 \cdot \mu_1(\sigma) + \cdots + p_k \cdot \mu_k(\sigma)$, $\mu_i \models \lceil b_i \rceil$ for all i such that $p_i > 0$, and $\sigma \models \neg(b_i \wedge b_j)$ for all σ, i, j such that $i \neq j$, then $\mu|_{b_i} = \mu_i$ for all i such that $[\![\boldsymbol{Pr}(b_i)]\!]_\mu > 0$.*

Proof. For all $\mu, \mu_1, \ldots, \mu_k, p_1, \ldots, p_k, b_1, \ldots, b_k$ such that $\mu = \lambda\sigma.\ p_1 \cdot \mu_1(\sigma) + \cdots + p_k \cdot \mu_k(\sigma)$, $\mu_i \models \lceil b_i \rceil$ for all i such that $p_i > 0$, and $\sigma \models \neg(b_i \wedge b_j)$ for all σ, i, j such that $i \neq j$. for all i such that $[\![\mathbf{Pr}(b_i)]\!]_\mu > 0$, by Lem. 230 we know $p_j = 0$ or $[\![\mathbf{Pr}(b_i)]\!]_{\mu_j} = 0$ for all j such that $i \neq j$, thus

$$
\begin{aligned}
&[\![\mathbf{Pr}(b_i)]\!]_\mu \\
&= \mathbf{Pr}_{\sigma\sim\mu}[\sigma \models b_i] \\
&= \textstyle\sum_\sigma\{\mu(\sigma) \mid \sigma \models b_i\} \\
&= \textstyle\sum_\sigma\{p_1 \cdot \mu_1(\sigma) + \cdots + p_k \cdot \mu_k(\sigma) \mid \sigma \models b_i\} \\
&= p_1 \cdot \textstyle\sum_\sigma\{\mu_1(\sigma) \mid \sigma \models b_i\} + \cdots + p_k \cdot \sum_\sigma\{\mu_k(\sigma) \mid \sigma \models b_i\} \\
&= p_1 \cdot [\![\mathbf{Pr}(b_i)]\!]_{\mu_1} + \cdots + p_k \cdot [\![\mathbf{Pr}(b_i)]\!]_{\mu_k} \\
&= p_i \cdot [\![\mathbf{Pr}(b_i)]\!]_{\mu_i}.
\end{aligned}
$$

From $[\![\mathbf{Pr}(b_i)]\!]_\mu > 0$ we know $p_i > 0$, thus $\mu_i \models \lceil b_i \rceil$. Therefore,

$$
\begin{aligned}
\mu|_{b_i} &= \lambda\sigma. \begin{cases} \frac{\mu(\sigma)}{\mathbf{Pr}_{\sigma\sim\mu}[\sigma\models b_i]}, & \text{if } \sigma \models b_i \\ 0, & \text{otherwise} \end{cases} \\
&= \lambda\sigma. \begin{cases} \frac{p_1\cdot\mu_1(\sigma)+\cdots+p_k\cdot\mu_k(\sigma)}{[\![\mathbf{Pr}(b_i)]\!]_\mu}, & \text{if } \sigma \models b_i \\ 0, & \text{otherwise} \end{cases} \\
&= \lambda\sigma. \begin{cases} \frac{p_i\cdot\mu_i(\sigma)}{p_i\cdot[\![\mathbf{Pr}(b_i)]\!]_{\mu_i}}, & \text{if } \sigma \models b_i \\ 0, & \text{otherwise} \end{cases} && \text{(by Lem. 229)} \\
&= \lambda\sigma. \begin{cases} \mu_i(\sigma), & \text{if } \sigma \models b_i \\ 0, & \text{otherwise} \end{cases} && \text{(by Lem. 226)} \\
&= \lambda\sigma. \begin{cases} \mu_i(\sigma), & \text{if } \sigma \models b_i \\ \mu_i(\sigma), & \text{otherwise} \end{cases} && \text{(by Lem. 228)} \\
&= \mu_i.
\end{aligned}
$$

Lemma 232 (Soundness of (ATOM-SPLIT) rule). *For all C, R, G, I, P, Q, if* $\mathbf{Sta}(P, R, I)$, $\mathbf{Sta}(Q, R, I)$, $G \models_{\text{SQ}} \{I \wedge P\}C\{(I \wedge Q \wedge \lceil b_1 \rceil) \oplus \cdots \oplus (I \wedge Q \wedge \lceil b_k \rceil)\}$ *and* $\mathbf{Id} \Rightarrow G$,
then $R, G, I \models_{\text{ST}} \{P\}\langle C\rangle$ ***split***$(b_1, \ldots, b_k)\{Q\}$.

Proof. For all C, R, G, I, P, Q such that $\mathbf{Sta}(P, R, I)$, $\mathbf{Sta}(Q, R, I)$, $G \models_{\text{SQ}} \{I \wedge P\}C\{(I \wedge Q \wedge \lceil b_1 \rceil) \oplus \cdots \oplus (I \wedge Q \wedge \lceil b_k \rceil)\}$ and $\mathbf{Id} \Rightarrow G$, to prove $R, G, I \models_{\text{ST}} \{P\}\langle C\rangle$ $\mathbf{split}(b_1, \ldots, b_k)\{Q\}$, we need to prove for all μ, if $\mu \models I \wedge P$, then $(\mathit{init}(\langle C\rangle\ \mathbf{split}(b_1, \ldots, b_k), \mu), R, I) \Longrightarrow^n_{\text{ST}} (G, Q)$ for all n. For all μ such that $\mu \models I \wedge P$, by Lem. 18 we know $\mathit{init}(\langle C\rangle\ \mathbf{split}(b_1, \ldots, b_k), \mu)^{(Stmt)} = (\delta(\langle C\rangle\ \mathbf{split}(b_1, \ldots, b_k)) \otimes \mu)^{(Stmt)} = \delta(\langle C\rangle\ \mathbf{split}(b_1, \ldots, b_k))$.
To prove $(\mathit{init}(\langle C\rangle\ \mathbf{split}(b_1, \ldots, b_k), \mu), R, I) \Longrightarrow^n_{\text{ST}} (G, Q)$ for all n, it suffices to prove for all n and η, if $\eta^{(Stmt)} = \delta(\langle C\rangle\ \mathbf{split}(b_1, \ldots, b_k))$ and $\eta^{(State)} \models I \wedge P$, then $(\eta, R, I) \Longrightarrow^n_{\text{ST}} (G, Q)$. We prove by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all η, if $\eta^{(Stmt)} = \delta(\langle C\rangle\ \mathbf{split}(b_1, \ldots, b_k))$ and $\eta^{(State)} \models I \wedge P$, then $(\eta, R, I) \Longrightarrow^k_{\text{ST}} (G, Q)$.
  For all η such that $\eta^{(Stmt)} = \delta(\langle C\rangle\ \mathbf{split}(b_1, \ldots, b_k))$ and $\eta^{(State)} \models I \wedge P$, we need to prove
  - $\eta^{(Stmt)}(\mathbf{skip}) = 1$ or $\eta^{(Stmt)}(\mathbf{skip}) = 0$.
    From $\eta^{(Stmt)} = \delta(\langle C\rangle\ \mathbf{split}(b_1, \ldots, b_k))$ we know
    $\eta^{(Stmt)}(\mathbf{skip}) = \delta(\langle C\rangle\ \mathbf{split}(b_1, \ldots, b_k))(\mathbf{skip}) = 0$.
  - if $\eta^{(Stmt)}(\mathbf{skip}) > 0$, then $\eta|_{\mathbf{skip}}{}^{(State)} \models Q$.
    $\eta^{(Stmt)}(\mathbf{skip}) > 0$ contradicts with $\eta^{(Stmt)}(\mathbf{skip}) = 0$.
  - $\eta^{(State)} \models I$.
    From $\eta^{(State)} \models I \wedge P$ we know $\eta^{(State)} \models I$.
  - for all η', if $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longrightarrow^k_{\square} (G, Q)$.
    For all η' such that $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, by Lem. 188 we know $\mathit{supp}(\eta'^{(Stmt)}) \subseteq$

$supp(\eta^{(Stmt)})$. From $\eta^{(Stmt)} = \delta(\langle C\rangle\ \mathbf{split}(b_1,\ldots,b_k))$ by Lem. 27 we know $\eta'^{(Stmt)} = \delta(\langle C\rangle\ \mathbf{split}(b_1,\ldots,b_k))$. From $\mathbf{Sta}(P,R,I)$, $\eta^{(State)} \models I \wedge P$ and $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$ by Lem. 186 we have $\eta'^{(State)} \models I \wedge P$. From $\eta'^{(Stmt)} = \delta(\langle C\rangle\ \mathbf{split}(b_1,\ldots,b_k))$ by IH we have $(\eta', R, I) \Longmapsto^k_{\mathrm{ST}} (G, Q)$.

- for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, $\eta'^{(State)} \models I$ and $(\eta', R, I) \Longmapsto^k_{\square} (G, Q)$.
  For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, from $\eta^{(Stmt)} = \delta(\langle C\rangle\ \mathbf{split}(b_1,\ldots,b_k))$ by Lem. 190 we have $nextsplit(\eta) = \{nextsplit(\langle C\rangle\ \mathbf{split}(b_1,\ldots,b_k))\} = \{\mathbf{split}(b_1,\ldots,b_k)\}$. From $\eta \hookrightarrow (\theta, \eta')$ we know there exists η'' and i such that $\eta \rightsquigarrow (\theta, \eta'')$ and $\eta''|_{b_i} = \eta'$. From $\eta^{(Stmt)} = \delta(\langle C\rangle\ \mathbf{split}(b_1,\ldots,b_k))$ by Lem. 222 we know $\eta \rightsquigarrow (\{(\sigma,\sigma') \mid \sigma \in supp(\eta^{(State)}) \wedge \sigma' \in supp([\![C]\!](\sigma))\}, \delta(\mathbf{skip}) \otimes [\![C]\!](\eta^{(State)}))$. From $\eta \rightsquigarrow (\theta, \eta'')$ by Lem. 193 we know $\theta = \{(\sigma,\sigma') \mid \sigma \in supp(\eta^{(State)}) \wedge \sigma' \in supp([\![C]\!](\sigma))\}$ and $\eta'' = \delta(\mathbf{skip}) \otimes [\![C]\!](\eta^{(State)})$. By Lem. 18 and Lem. 19 we know $\eta''^{(Stmt)} = \delta(\mathbf{skip})$ and $\eta''^{(State)} = [\![C]\!](\eta^{(State)})$. From $|\eta''^{(State)}| = 1$ we know $|[\![C]\!](\eta^{(State)})| = 1$. From $\models_{\mathrm{SQ}} \{I \wedge P\}C\{(I \wedge Q \wedge \lceil b_1\rceil) \oplus \cdots \oplus (I \wedge Q \wedge \lceil b_k\rceil)\}$ and $\eta^{(State)} \models I \wedge P$ we know $[\![C]\!](\eta^{(State)}) \models (I \wedge Q \wedge \lceil b_1\rceil) \oplus \cdots \oplus (I \wedge Q \wedge \lceil b_k\rceil)$ and $(\sigma,\sigma') \models G$ for all $\sigma \in supp(\eta^{(State)})$ and $\sigma' \in supp([\![C]\!](\sigma))$, thus $\theta = \{(\sigma,\sigma') \mid \sigma \in supp(\eta^{(State)}) \wedge \sigma' \in supp([\![C]\!](\sigma))\} \subseteq [\![G]\!]$ and $\eta''^{(State)} = [\![C]\!](\eta^{(State)}) \models (I \wedge Q \wedge \lceil b_1\rceil) \oplus \cdots \oplus (I \wedge Q \wedge \lceil b_k\rceil)$, By Lem. 225 we know so there exists $\mu_1,\ldots,\mu_k, p_1,\ldots,p_k$ such that $\eta''^{(State)} = \lambda\sigma.\ p_1 \cdot \mu_1(\sigma) + \cdots + p_k \cdot \mu_k(\sigma)$ and $\mu_j \models I \wedge Q \wedge \lceil b_j\rceil$ for all j. From $\eta''|_{b_i} = \eta'$ by Lem. 205 we know $[\![\mathbf{Pr}(b_i)]\!]_{\eta''^{(State)}} > 0$. By Lem. 206 we know $\eta'^{(State)} = \eta''|_{b_i}{}^{(State)} = \eta''^{(State)}|_{b_i}$. From $\mathbf{validsplit}(\mathbf{split}(b_1,\ldots,b_k))$ we know $\sigma \models \neg(b_i \wedge b_j)$ for all σ, i, j such that $i \neq j$. From $\eta''^{(State)} = \lambda\sigma.\ p_1 \cdot \mu_1(\sigma) + \cdots + p_k \cdot \mu_k(\sigma)$, $\mu_j \models \lceil b_j\rceil$ for all j, and $[\![\mathbf{Pr}(b_i)]\!]_{\eta''^{(State)}} > 0$ by Lem. 231 we know $\eta''^{(State)}|_{b_i} = \mu_i$, thus $\eta'^{(State)} = \eta''^{(State)}|_{b_i} = \mu_i \models I \wedge Q \wedge \lceil b_i\rceil$. From $\eta''|_{b_i} = \eta'$ by Lem. 20 we know $supp(\eta') \subseteq supp(\eta'')$, thus $dom(supp(\eta')) \subseteq dom(supp(\eta''))$. By Lem. 21 we know $supp(\eta'^{(Stmt)}) = dom(supp(\eta')) \subseteq dom(supp(\eta'')) = supp(\eta''^{(Stmt)})$. From $\eta''^{(Stmt)} = \delta(\mathbf{skip})$ by Lem. 27 we know $\eta'^{(Stmt)} = \delta(\mathbf{skip})$. From $\mathbf{Sta}(Q,R,I)$, $\mathbf{Id} \Rightarrow G$, $\eta'^{(Stmt)} = \delta(\mathbf{skip})$ and $\eta'^{(State)} \models I \wedge Q$ by Lem. 194 we know $(\eta', R, I) \Longmapsto^k_{\mathrm{ST}} (G, Q)$.

Lemma 233. *For all C, R, G, I, P, Q, if $R, G, I \vdash_{ST} \{P\}C\{Q\}$, then $R, G, I \models_{ST} \{P\}C\{Q\}$.*

Proof. For all C, R, G, I, P, Q such that $R, G, I \vdash_{\mathrm{ST}} \{P\}C\{Q\}$, we prove $R, G, I \models_{\mathrm{ST}} \{P\}C\{Q\}$ by induction on the derivation of $R, G, I \vdash_{\mathrm{ST}} \{P\}C\{Q\}$.

- case (DISJ): $P = P_1 \vee P_2$, $Q = Q_1 \vee Q_2$, $R, G, I \vdash_{\mathrm{ST}} \{P_1\}C\{Q_1\}$ and $R, G, I \vdash_{\mathrm{ST}} \{P_2\}C\{Q_2\}$.
  From $R, G, I \vdash_{\mathrm{ST}} \{P_1\}C\{Q_1\}$ by induction hypothesis we know $R, G, I \models_{\mathrm{ST}} \{P_1\}C\{Q_1\}$. From $R, G, I \vdash_{\mathrm{ST}} \{P_2\}C\{Q_2\}$ by induction hypothesis we know

$R, G, I \models_{\text{ST}} \{P_2\}C\{Q_2\}$. By Lem. 183 we know $R, G, I \vdash_{\text{ST}} \{P_1 \vee P_2\}C\{Q_1 \vee Q_2\}$, i.e., $R, G, I \vdash_{\text{ST}} \{P\}C\{Q\}$.

- case (CONJ): $P = P_1 \wedge P_2$, $Q = Q_1 \wedge Q_2$, $R, G, I \vdash_{\text{ST}} \{P_1\}C\{Q_1\}$ and $R, G, I \vdash_{\text{ST}} \{P_2\}C\{Q_2\}$.
  From $R, G, I \vdash_{\text{ST}} \{P_1\}C\{Q_1\}$ by induction hypothesis we know $R, G, I \models_{\text{ST}} \{P_1\}C\{Q_1\}$. From $R, G, I \vdash_{\text{ST}} \{P_2\}C\{Q_2\}$ by induction hypothesis we know $R, G, I \models_{\text{ST}} \{P_2\}C\{Q_2\}$. By Lem. 185 we know $R, G, I \vdash_{\text{ST}} \{P_1 \wedge P_2\}C\{Q_1 \wedge Q_2\}$, i.e., $R, G, I \vdash_{\text{ST}} \{P\}C\{Q\}$.
- case (CSQ): $P \Rightarrow P_1$, $R \Rightarrow R_1$, $G_1 \Rightarrow G$, $Q_1 \Rightarrow Q$ and $R_1, G_1, I \vdash_{\text{ST}} \{P_1\}C\{Q_1\}$.
  From $R_1, G_1, I \vdash_{\text{ST}} \{P_1\}C\{Q_1\}$ by induction hypothesis we know $R_1, G_1, I \models_{\text{ST}} \{P_1\}C\{Q_1\}$. From $P \Rightarrow P_1$, $R \Rightarrow R_1$, $G_1 \Rightarrow G$ and $Q_1 \Rightarrow Q$ by Lem. 182 we know $R, G, I \models_{\text{ST}} \{P\}C\{Q\}$.
- case (SKIP): $P = Q$, $G = \mathbf{Id}$, $\mathbf{Sta}(Q, R, I)$.
  From $\mathbf{Sta}(Q, R, I)$ and $G = \mathbf{Id}$ by Lem. 195 we have $R, G, I \models_{\text{ST}} \{Q\}\mathbf{skip}\{Q\}$, i.e., $R, G, I \models_{\text{ST}} \{P\}\mathbf{skip}\{Q\}$.
- case (ATOM): $C = \langle C_1 \rangle$, $\mathbf{Sta}(P, R, I)$, $\mathbf{Sta}(Q, R, I)$ and $G \vdash_{\text{SQ}} \{I \wedge P\}C_1\{I \wedge Q\}$.
  From $G \vdash_{\text{SQ}} \{I \wedge P\}C_1\{I \wedge Q\}$ by Lem. 367 we know $G \models_{\text{SQ}} \{I \wedge P\}C_1\{I \wedge Q\}$. From $\mathbf{Sta}(P, R, I)$ and $\mathbf{Sta}(Q, R, I)$ by Lem. 223 we know $R, G, I \vdash_{\text{SQ}} \{P\}\langle C_1 \rangle\{Q\}$, i.e., $R, G, I \vdash_{\text{SQ}} \{P\}C\{Q\}$.
- case (ATOM-SPLIT): $C = \langle C_1 \rangle\ \mathbf{split}(b_1, \ldots, b_k)$, $\mathbf{Sta}(P, R, I)$, $\mathbf{Sta}(Q, R, I)$ and $G \vdash_{\text{SQ}} \{I \wedge P\}C_1\{(I \wedge Q \wedge \lceil b_1 \rceil) \oplus \cdots \oplus (I \wedge Q \wedge \lceil b_k \rceil)\}$.
  From $G \vdash_{\text{SQ}} \{I \wedge P\}C_1\{(I \wedge Q \wedge \lceil b_1 \rceil) \oplus \cdots \oplus (I \wedge Q \wedge \lceil b_k \rceil)\}$ by Lem. 367 we know $G \models_{\text{SQ}} \{I \wedge P\}C_1\{(I \wedge Q \wedge \lceil b_1 \rceil) \oplus \cdots \oplus (I \wedge Q \wedge \lceil b_k \rceil)\}$. From $\mathbf{Sta}(P, R, I)$ and $\mathbf{Sta}(Q, R, I)$ by Lem. 232 we know $R, G, I \vdash_{\text{SQ}} \{P\}\langle C_1 \rangle\ \mathbf{split}(b_1, \ldots, b_k)\{Q\}$, i.e., $R, G, I \vdash_{\text{SQ}} \{P\}C\{Q\}$.
- case (SEQ-ST): $C = C_1; C_2$, $R, G, I \vdash_{\text{ST}} \{P\}C_1\{M\}$ and $R, G, I \vdash_{\text{ST}} \{M\}C_2\{Q\}$.
  From $R, G, I \vdash_{\text{ST}} \{P\}C_1\{M\}$ by induction hypothesis we have $R, G, I \models_{\text{ST}} \{P\}C_1\{M\}$. From $R, G, I \vdash_{\text{ST}} \{M\}C_1\{Q\}$ by induction hypothesis we have $R, G, I \models_{\text{ST}} \{M\}C_1\{Q\}$. By Lem. 218 we know $R, G, I \models_{\text{ST}} \{P\}C_1; C_2\{Q\}$, i.e., $R, G, I \models_{\text{ST}} \{P\}C\{Q\}$.
- case (COND): $C = \mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2$, $P = P_1 \vee P_2$, $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $P_1 \Rightarrow \lceil b \rceil$, $P_2 \Rightarrow \lceil \neg b \rceil$, $R, G, I \vdash_{\text{ST}} \{P_1\}C_1\{Q\}$ and $R, G, I \vdash_{\text{ST}} \{P_2\}C_1\{Q\}$.
  From $R, G, I \vdash_{\text{ST}} \{P_1\}C_1\{Q\}$ by induction hypothesis we know $R, G, I \models_{\text{ST}} \{P_1\}C_1\{Q\}$. From $R, G, I \vdash_{\text{ST}} \{P_2\}C_1\{Q\}$ by induction hypothesis we know $R, G, I \models_{\text{ST}} \{P_2\}C_2\{Q\}$. From $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $P_1 \Rightarrow \lceil b \rceil$ and $P_2 \Rightarrow \lceil \neg b \rceil$ by Lem. 200 we know $R, G, I \models_{\text{ST}} \{P_1 \vee P_2\}\mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2\{Q\}$, i.e., $R, G, I \models_{\text{ST}} \{P\}C\{Q\}$.
- case (WHILE-ST): $C = \mathbf{while}\ (b)\ \mathbf{do}\ C_1$, $P = P_1 \vee P_2$, $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $\mathbf{Sta}(Q, R, I)$, $P_1 \Rightarrow \lceil b \rceil$, $P_2 \Rightarrow \lceil \neg b \rceil \wedge Q$, $R, G, I \vdash_{\text{ST}} \{P_1\}C_1\{P_1 \vee P_2\}$.
  From $R, G, I \vdash_{\text{ST}} \{P_1\}C_1\{P_1 \vee P_2\}$ by induction hypothesis we know $R, G, I \models_{\text{ST}} \{P_1\}C_1\{P_1 \vee P_2\}$. From $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $\mathbf{Sta}(Q, R, I)$, $P_1 \Rightarrow \lceil b \rceil$ and $P_2 \Rightarrow \lceil \neg b \rceil \wedge Q$ by Lem. 221 we know $R, G, I \models_{\text{ST}} \{P_1 \vee P_2\}\mathbf{while}\ (b)\ \mathbf{do}\ C_1\{Q\}$, i.e., $R, G, I \models_{\text{ST}} \{P\}C\{Q\}$.

Lemma 234. *For all η_1, η_2, p, if ${\eta_1}^{(Stmt)}(\boldsymbol{skip}) = 0$ and $(\eta_1 \oplus_p \eta_2)^{(Stmt)}(\boldsymbol{skip}) > 0$,*
then $(\eta_1 \oplus_p \eta_2)|_{\boldsymbol{skip}} = \eta_2|_{\boldsymbol{skip}}$.

Proof. For all η_1, η_2, p such that ${\eta_1}^{(Stmt)}(\mathbf{skip}) = 0$ and $(\eta_1 \oplus_p \eta_2)^{(Stmt)}(\mathbf{skip}) > 0$,
from ${\eta_1}^{(Stmt)}(\mathbf{skip}) = 0$ we know $\sum_\sigma \eta_1(\mathbf{skip}, \sigma) = 0$, so $\eta_1(\mathbf{skip}, \sigma) = 0$ for all σ, thus

$$
\begin{aligned}
& (\eta_1 \oplus_p \eta_2)|_{\mathbf{skip}} \\
&= \lambda(C,\sigma).\frac{\chi(C{=}\mathbf{skip})\cdot(\eta_1\oplus_p\eta_2)(\mathbf{skip},\sigma)}{(\eta_1\oplus_p\eta_2)^{(Stmt)}(\mathbf{skip})} && \text{(by Lem. 165)} \\
&= \lambda(C,\sigma).\frac{\chi(C{=}\mathbf{skip})\cdot(\eta_1\oplus_p\eta_2)(\mathbf{skip},\sigma)}{({\eta_1}^{(Stmt)}\oplus_p{\eta_2}^{(Stmt)})(\mathbf{skip})} && \text{(by Lem. 11)} \\
&= \lambda(C,\sigma).\frac{\chi(C{=}\mathbf{skip})\cdot(p\cdot\eta_1(\mathbf{skip},\sigma)+(1-p)\cdot\eta_2(\mathbf{skip},\sigma))}{p\cdot{\eta_1}^{(Stmt)}(\mathbf{skip})+(1-p)\cdot{\eta_2}^{(Stmt)}(\mathbf{skip})} \\
&= \lambda(C,\sigma).\frac{\chi(C{=}\mathbf{skip})\cdot\eta_2(\mathbf{skip},\sigma)}{{\eta_2}^{(Stmt)}(\mathbf{skip})} \\
&= \eta|_{\mathbf{skip}}. && \text{(by Lem. 165)}
\end{aligned}
$$

Lemma 235. *For all $\eta_1, \eta_2, p, R, \eta'$, if $0 < p < 1$ and $\eta_1 \oplus_p \eta_2 \overset{R}{\mapsto} \eta'$, then there exists η'_1, η'_2, p' such that $0 < p' < 1$, $\eta = \eta'_1 \oplus_{p'} \eta'_2$, $\eta_1 \overset{R}{\mapsto} \eta'_1$ and $\eta_2 \overset{R}{\mapsto} \eta'_2$.*

Proof. For all $\eta_1, \eta_2, p, R, \eta'$ such that $0 < p < 1$ and $\eta_1 \oplus_p \eta_2 \overset{R}{\mapsto} \eta'$, there exists ψ such that $dom(\psi) = supp(\eta_1 \oplus_p \eta_2)$, $range(\psi) = supp(\eta')$ and $\forall((C,\sigma),(C',\sigma')) \in \psi.\ C' = C \wedge (\sigma,\sigma') \models R$. Let $\psi_1 \overset{\text{def}}{=} \{((C,\sigma),(C',\sigma')) \mid (C,\sigma) \in supp(\eta_1) \wedge ((C,\sigma),(C',\sigma')) \in \psi\}$, $\psi_2 \overset{\text{def}}{=} \{((C,\sigma),(C',\sigma')) \mid (C,\sigma) \in supp(\eta_2) \wedge ((C,\sigma),(C',\sigma')) \in \psi\}$, $p' \overset{\text{def}}{=} 0.5$, and

$$
\eta'_1 \overset{\text{def}}{=} \lambda(C,\sigma).\begin{cases} \eta'(C,\sigma), & \text{if } (C,\sigma) \in range(\psi_1) \cap range(\psi_2) \\ 2\eta'(C,\sigma), & \text{if } (C,\sigma) \in range(\psi_1) - range(\psi_2) \\ 0, & \text{otherwise} \end{cases}
$$

$$
\eta'_2 \overset{\text{def}}{=} \lambda(C,\sigma).\begin{cases} \eta'(C,\sigma), & \text{if } (C,\sigma) \in range(\psi_1) \cap range(\psi_2) \\ 2\eta'(C,\sigma), & \text{if } (C,\sigma) \in range(\psi_2) - range(\psi_1) \\ 0, & \text{otherwise} \end{cases}
$$

From $0 < p < 1$ by Lem. 275 we know $supp(\eta) = supp(\eta_1 \oplus_p \eta_2) = supp(\eta_1) \cup supp(\eta_2)$, thus

$$
\begin{aligned}
& \psi_1 \cup \psi_2 \\
&= \{((C,\sigma),(C',\sigma')) \mid (C,\sigma) \in supp(\eta_1) \wedge ((C,\sigma),(C',\sigma')) \in \psi\} \cup \\
&\quad \{((C,\sigma),(C',\sigma')) \mid (C,\sigma) \in supp(\eta_2) \wedge ((C,\sigma),(C',\sigma')) \in \psi\} \\
&= \{((C,\sigma),(C',\sigma')) \mid ((C,\sigma) \in supp(\eta_1) \vee (C,\sigma) \in supp(\eta_1)) \wedge ((C,\sigma),(C',\sigma')) \in \psi\} \\
&= \{((C,\sigma),(C',\sigma')) \mid ((C,\sigma) \in supp(\eta_1) \cup supp(\eta_2)) \wedge ((C,\sigma),(C',\sigma')) \in \psi\} \\
&= \{((C,\sigma),(C',\sigma')) \mid (C,\sigma) \in supp(\eta) \wedge ((C,\sigma),(C',\sigma')) \in \psi\} \\
&= \{((C,\sigma),(C',\sigma')) \mid (C,\sigma) \in dom(\psi) \wedge ((C,\sigma),(C',\sigma')) \in \psi\} \\
&= \{((C,\sigma),(C',\sigma')) \mid ((C,\sigma),(C',\sigma')) \in \psi\} \\
&= \psi,
\end{aligned}
$$

so $range(\psi_1) \cup range(\psi_2) = range(\psi_1 \cup \psi_2) = range(\psi)$. Therefore,

$$\begin{aligned}
\eta_1' \oplus_p \eta_2' &= \lambda(C,\sigma).\ p' \cdot \eta_1'(C,\sigma) + (1-p') \cdot \eta_2'(C,\sigma) \\
&= \lambda(C,\sigma).\ 0.5 \cdot \eta_1'(C,\sigma) + 0.5 \cdot \eta_2'(C,\sigma) \\
&= \lambda(C,\sigma).\ \begin{cases} \eta'(C,\sigma), & \text{if } (C,\sigma) \in range(\psi_1) \cap range(\psi_2) \\ \eta'(C,\sigma), & \text{if } (C,\sigma) \in range(\psi_1) - range(\psi_2) \\ \eta'(C,\sigma), & \text{if } (C,\sigma) \in range(\psi_2) - range(\psi_2) \\ 0, & \text{if } (C,\sigma) \notin range(\psi_1) \cup range(\psi_2) \end{cases} \\
&= \lambda(C,\sigma).\ \begin{cases} \eta'(C,\sigma), & \text{if } (C,\sigma) \in range(\psi) \\ 0, & \text{if } (C,\sigma) \notin range(\psi) \end{cases} \\
&= \lambda(C,\sigma).\ \begin{cases} \eta'(C,\sigma), & \text{if } (C,\sigma) \in supp(\eta') \\ 0, & \text{if } (C,\sigma) \notin supp(\eta') \end{cases} \\
&= \eta'.
\end{aligned}$$

$$\begin{aligned}
dom(\psi_1) &= \{(C,\sigma) \mid \exists C', \sigma'.\ ((C,\sigma),(C',\sigma')) \in \psi_1\} \\
&= \{(C,\sigma) \mid (C,\sigma) \in supp(\eta_1) \wedge \exists C', \sigma'.\ ((C,\sigma),(C',\sigma')) \in \psi\} \\
&= \{(C,\sigma) \mid (C,\sigma) \in supp(\eta_1) \wedge (C,\sigma) \in dom(\psi)\} \\
&= \{(C,\sigma) \mid (C,\sigma) \in supp(\eta_1) \wedge (C,\sigma) \in supp(\eta)\} \\
&= \{(C,\sigma) \mid (C,\sigma) \in supp(\eta_1) \cap supp(\eta)\} \\
&= \{(C,\sigma) \mid (C,\sigma) \in supp(\eta_1) \cap (supp(\eta_1) \cup supp(\eta_2))\} \\
&= \{(C,\sigma) \mid (C,\sigma) \in supp(\eta_1)\} \\
&= supp(\eta_1).
\end{aligned}$$

$$\begin{aligned}
&supp(\eta_1') \\
&= \{(C,\sigma) \mid \eta_1'(C,\sigma) > 0\} \\
&= \{(C,\sigma) \mid (\eta'(C,\sigma) > 0 \wedge (C,\sigma) \in range(\psi_1) \cap range(\psi_2)) \vee \\
&\qquad (2 \cdot \eta'(C,\sigma) > 0 \wedge (C,\sigma) \in range(\psi_1) - range(\psi_2))\} \\
&= \{(C,\sigma) \mid \eta'(C,\sigma) > 0 \wedge ((C,\sigma) \in range(\psi_1) \cap range(\psi_2) \vee (C,\sigma) \in range(\psi_1) - range(\psi_2))\} \\
&= \{(C,\sigma) \mid \eta'(C,\sigma) > 0 \wedge (C,\sigma) \in (range(\psi_1) \cap range(\psi_2)) \cup (range(\psi_1) - range(\psi_2))\} \\
&= \{(C,\sigma) \mid (C,\sigma) \in supp(\eta') \wedge (C,\sigma) \in range(\psi_1)\} \\
&= supp(\eta') \cap range(\psi_1) \\
&= range(\psi) \cap range(\psi_1) \\
&= (range(\psi_1) \cup range(\psi_2)) \cap range(\psi_1) \\
&= range(\psi_1).
\end{aligned}$$

From $\psi = \psi_1 \cup \psi_2 \supseteq \psi_1$ and $\forall((C,\sigma),(C',\sigma')) \in \psi.\ C' = C \wedge (\sigma,\sigma') \models R$ we know $\forall((C,\sigma),(C',\sigma')) \in \psi_1.\ C' = C \wedge (\sigma,\sigma') \models R$. From $dom(\psi_1) = supp(\eta_1)$ and $range(\psi_1) = supp(\eta_1')$ we know $\eta_1 \overset{R}{\mapsto} \eta_1'$. Similarly, we can prove $\eta_2 \overset{R}{\mapsto} \eta_2'$.

Lemma 236. *For all $\eta_1, \eta_2, p, R, \eta_1', \eta_2', p'$, if $0 < p < 1$, $\eta_1 \overset{R}{\mapsto} \eta_1'$, $\eta_2 \overset{R}{\mapsto} \eta_2'$ and $0 < p' < 1$, then $\eta_1 \oplus_p \eta_2 \overset{R}{\mapsto} \eta_1' \oplus_{p'} \eta_2'$.*

Proof. For all $\eta_1, \eta_2, p, R, \eta_1', \eta_2', p'$ such that $0 < p < 1$, $\eta_1 \overset{R}{\mapsto} \eta_1'$, $\eta_2 \overset{R}{\mapsto} \eta_2'$ and $0 < p' < 1$, from $\eta_1 \overset{R}{\mapsto} \eta_1'$ we know there exists ψ_1 such that $dom(\psi_1) = supp(\eta_1)$,

$range(\psi_1) = supp(\eta_1')$ and $\forall((C,\sigma),(C',\sigma')) \in \psi_1.\ C' = C \wedge (\sigma,\sigma') \models R$. From $\eta_2 \overset{R}{\mapsto} \eta_2'$ we know there exists ψ_2 such that $dom(\psi_2) = supp(\eta_2)$, $range(\psi_2) = supp(\eta_2')$ and $\forall((C,\sigma),(C',\sigma')) \in \psi_2.\ C' = C \wedge (\sigma,\sigma') \models R$. Let $\psi = \psi_1 \cup \psi_2$, then $dom(\psi) = dom(\psi_1) \cup dom(\psi_2) = supp(\eta_1) \cup supp(\eta_2)$, $range(\psi) = range(\psi_1) \cup range(\psi_2) = supp(\eta_1') \cup supp(\eta_2')$ and $\forall((C,\sigma),(C',\sigma')) \in \psi.\ C' = C \wedge (\sigma,\sigma') \models R$. From $0 < p < 1$ and $0 < p' < 1$ by Lem. 275 we know $supp(\eta_1 \oplus_p \eta_2) = supp(\eta_1) \cup supp(\eta_2)$ and $supp(\eta_1' \oplus_{p'} \eta_2') = supp(\eta_1') \cup supp(\eta_2')$. thus $dom(\psi) = supp(\eta_1 \oplus_p \eta_1)$ and $range(\psi) = supp(\eta_1' \oplus_{p'} \eta_2')$. Therefore, $\eta_1 \oplus_p \eta_2 \overset{R}{\mapsto} \eta_1' \oplus_{p'} \eta_2'$.

Lemma 237. *For all* ξ, μ_1, μ_2, p, $[\![\xi]\!]_{\mu_1 \oplus_p \mu_2} = p \cdot [\![\xi]\!]_{\mu_1} + (1-p) \cdot [\![\xi]\!]_{\mu_2}$.

Proof. For all ξ, μ_1, μ_2, p, we prove $[\![\xi]\!]_{\mu_1 \oplus_p \mu_2} = p \cdot [\![\xi]\!]_{\mu_1} + (1-p) \cdot [\![\xi]\!]_{\mu_2}$ by induction on the structure of ξ.

- case $\xi = r$.
  $[\![\xi]\!]_{\mu_1 \oplus_p \mu_2} = [\![r]\!]_{\mu_1 \oplus_p \mu_2} = r = p \cdot r + (1-p) \cdot r = p \cdot [\![r]\!]_{\mu_1} + (1-p) \cdot [\![r]\!]_{\mu_2} = [\![\xi]\!]_{\mu_1} + (1-p) \cdot [\![\xi]\!]_{\mu_2}$.
- case $\xi = \mathbb{E}(e)$.

$$\begin{aligned}
[\![\xi]\!]_{\mu_1 \oplus_p \mu_2} &= [\![\mathbb{E}(e)]\!]_{\mu_1 \oplus_p \mu_2} \\
&= \mathbb{E}_{\sigma \sim \mu_1 \oplus_p \mu_2}[[\![e]\!]_\sigma] \\
&= p \cdot \mathbb{E}_{\sigma \sim \mu_1}[[\![e]\!]_\sigma] + (1-p) \cdot \mathbb{E}_{\sigma \sim \mu_2}[[\![e]\!]_\sigma] \qquad \text{(by Lem. 16)} \\
&= p \cdot [\![\mathbb{E}(e)]\!]_{\mu_1} + (1-p) \cdot [\![\mathbb{E}(e)]\!]_{\mu_2} \\
&= p \cdot [\![\xi]\!]_{\mu_1} + (1-p) \cdot [\![\xi]\!]_{\mu_2}.
\end{aligned}$$

- case $\xi = \mathbf{Pr}(\mathbf{q})$.

$$\begin{aligned}
[\![\xi]\!]_{\mu_1 \oplus_p \mu_2} &= [\![\mathbf{Pr}(\mathbf{q})]\!]_{\mu_1 \oplus_p \mu_2} \\
&= \mathbf{Pr}_{\sigma \sim \mu_1 \oplus_p \mu_2}[\sigma \models \mathbf{q}] \\
&= \textstyle\sum_\sigma \{(\mu_1 \oplus_p \mu_2)(\sigma) \mid \sigma \models \mathbf{q}\} \\
&= \textstyle\sum_\sigma \{p \cdot \mu_1(\sigma) + (1-p) \cdot \mu_2(\sigma) \mid \sigma \models \mathbf{q}\} \\
&= p \cdot \textstyle\sum_\sigma \{\mu_1(\sigma) \mid \sigma \models \mathbf{q}\} + (1-p) \cdot \textstyle\sum_\sigma \{\mu_2(\sigma) \mid \sigma \models \mathbf{q}\} \\
&= p \cdot \mathbf{Pr}_{\sigma \sim \mu_1}[\sigma \models \mathbf{q}] + (1-p) \cdot \mathbf{Pr}_{\sigma \sim \mu_2}[\sigma \models \mathbf{q}] \\
&= p \cdot [\![\mathbf{Pr}(\mathbf{q})]\!]_{\mu_1} + (1-p) \cdot [\![\mathbf{Pr}(\mathbf{q})]\!]_{\mu_2} \\
&= p \cdot [\![\xi]\!]_{\mu_1} + (1-p) \cdot [\![\xi]\!]_{\mu_2}.
\end{aligned}$$

- case $\xi = \xi_1 + \xi_2$.

$$\begin{aligned}
&[\![\xi]\!]_{\mu_1 \oplus_p \mu_2} \\
&= [\![\xi_1 + \xi_2]\!]_{\mu_1 \oplus_p \mu_2} \\
&= [\![\xi_1]\!]_{\mu_1 \oplus_p \mu_2} + [\![\xi_2]\!]_{\mu_1 \oplus_p \mu_2} \\
&= p \cdot [\![\xi_1]\!]_{\mu_1} + (1-p) \cdot [\![\xi_1]\!]_{\mu_2} + p \cdot [\![\xi_2]\!]_{\mu_1} + (1-p) \cdot [\![\xi_2]\!]_{\mu_2} \qquad \text{(by induction hypothesis)} \\
&= p \cdot ([\![\xi_1]\!]_{\mu_1} + [\![\xi_2]\!]_{\mu_1}) + (1-p) \cdot ([\![\xi_1]\!]_{\mu_2} + [\![\xi_2]\!]_{\mu_2}) \\
&= p \cdot [\![\xi_1 + \xi_2]\!]_{\mu_1} + (1-p) \cdot [\![\xi_1 + \xi_2]\!]_{\mu_2} \\
&= p \cdot [\![\xi]\!]_{\mu_1} + (1-p) \cdot [\![\xi]\!]_{\mu_2}.
\end{aligned}$$

  Similarly, we can prove the case $\xi = \xi_1 - \xi_2$ and the case $\xi = \xi_1 * \xi_2$.

Lemma 238. *For all* η_1, η_2, p, b, *if* $0 < p < 1$, $[\![\boldsymbol{Pr}(b)]\!]_{\eta_1{}^{(State)}} = p_1$, *and* $[\![\boldsymbol{Pr}(b)]\!]_{\eta_2{}^{(State)}} = p_2$, *then* $(\eta_1 \oplus_p \eta_2)|_b = \begin{cases} \eta_1|_b \oplus_{\frac{p\cdot p_1}{p\cdot p_1+(1-p)\cdot p_2}} \eta_2|_b, & \text{if } p_1 > 0 \wedge p_2 > 0 \\ \eta_1|_b, & \text{if } p_1 > 0 \wedge p_2 = 0 \\ \eta_2|_b, & \text{if } p_1 = 0 \wedge p_2 > 0 \\ \text{undefined}, & \text{otherwise.} \end{cases}$

Proof. For all η_1, η_2, p, b such that $0 < p < 1$, $[\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}} = p_1$, and $[\![\mathbf{Pr}(b)]\!]_{\eta_2{}^{(State)}} = p_2$, we prove the four cases respectively.

- $p_1 > 0 \wedge p_2 > 0$.
  By Lem. 12 and Lem. 237 we know $[\![\mathbf{Pr}(b)]\!]_{(\eta_1\oplus_p\eta_2)^{(State)}} = [\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}\oplus_p\eta_2{}^{(State)}} = p \cdot [\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}} + (1-p) \cdot [\![\mathbf{Pr}(b)]\!]_{\eta_2{}^{(State)}} = p \cdot p_1 + (1-p) \cdot p_2 > 0$, thus

$$
\begin{aligned}
&(\eta_1 \oplus_p \eta_2)|_b \\
&= \lambda(C,\sigma).\ \tfrac{\chi(\sigma\models b)\cdot(\eta_1\oplus_p\eta_2)(C,\sigma)}{[\![\mathbf{Pr}(b)]\!]_{(\eta_1\oplus_p\eta_2)^{(State)}}} \qquad \text{(by Lem. 207)} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(\sigma\models b)\cdot(p\cdot\eta_1(C,\sigma)+(1-p)\cdot\eta_2(C,\sigma))}{p\cdot p_1+(1-p)\cdot p_2} \\
&= \lambda(C,\sigma).\ \tfrac{p\cdot p_1}{p\cdot p_1+(1-p)\cdot p_2}\cdot\tfrac{\chi(\sigma\models b)\cdot\eta_1(C,\sigma)}{p_1} + \tfrac{(1-p)\cdot p_2}{p\cdot p_1+(1-p)\cdot p_2}\cdot\tfrac{\chi(\sigma\models b)\cdot\eta_2(C,\sigma)}{p_2} \\
&= \lambda(C,\sigma).\ \tfrac{p\cdot p_1}{p\cdot p_1+(1-p)\cdot p_2}\cdot\tfrac{\chi(\sigma\models b)\cdot\eta_1(C,\sigma)}{[\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}}} + (1-\tfrac{p\cdot p_1}{p\cdot p_1+(1-p)\cdot p_2})\cdot\tfrac{\chi(\sigma\models b)\cdot\eta_2(C,\sigma)}{[\![\mathbf{Pr}(b)]\!]_{\eta_2{}^{(State)}}} \\
&= \lambda(C,\sigma).\ \tfrac{p\cdot p_1}{p\cdot p_1+(1-p)\cdot p_2}\cdot\eta_1|_b(C,\sigma) + (1-\tfrac{p\cdot p_1}{p\cdot p_1+(1-p)\cdot p_2})\cdot\eta_1|_b(C,\sigma) \qquad \text{(by Lem. 207)} \\
&= \eta_1|_b \oplus_{\frac{p\cdot p_1}{p\cdot p_1+(1-p)\cdot p_2}} \eta_2|_b.
\end{aligned}
$$

- $p_1 > 0 \wedge p_2 = 0$.
  By Lem. 12 and Lem. 237 we know $[\![\mathbf{Pr}(b)]\!]_{(\eta_1\oplus_p\eta_2)^{(State)}} = [\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}\oplus_p\eta_2{}^{(State)}} = p \cdot [\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}} + (1-p) \cdot [\![\mathbf{Pr}(b)]\!]_{\eta_2{}^{(State)}} = p \cdot p_1 + (1-p) \cdot p_2 = p \cdot p_1 > 0$.
  From $0 = p_2 = [\![\mathbf{Pr}(b)]\!]_{\eta_2{}^{(State)}} = \mathbf{Pr}_{\sigma\sim\eta_2{}^{(State)}}[\sigma \models b] = \mathbf{Pr}_{(C,\sigma)\sim\eta_2}[\sigma \models b] = \sum_{C,\sigma}\{\eta_2(C,\sigma) \mid \sigma \models b\}$ we know $\sigma \not\models b$ for all C and σ such that $\eta_2(C,\sigma) > 0$, thus

$$
\begin{aligned}
&(\eta_1 \oplus_p \eta_2)|_b \\
&= \lambda(C,\sigma).\ \tfrac{\chi(\sigma\models b)\cdot(\eta_1\oplus_p\eta_2)(C,\sigma)}{[\![\mathbf{Pr}(b)]\!]_{(\eta_1\oplus_p\eta_2)^{(State)}}} \qquad \text{(by Lem. 207)} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(\sigma\models b)\cdot(p\cdot\eta_1(C,\sigma)+(1-p)\cdot\eta_2(C,\sigma))}{p\cdot p_1} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(\sigma\models b)\cdot p\cdot\eta_1(C,\sigma)}{p\cdot p_1} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(\sigma\models b)\cdot\eta_1(C,\sigma)}{[\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}}} \\
&= \eta_1|_b. \qquad \text{(by Lem. 207)}
\end{aligned}
$$

- $p_1 = 0 \wedge p_2 > 0$.
  By Lem. 12 and Lem. 237 we know $[\![\mathbf{Pr}(b)]\!]_{(\eta_1\oplus_p\eta_2)^{(State)}} = [\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}\oplus_p\eta_2{}^{(State)}} = p\cdot[\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}} + (1-p)\cdot[\![\mathbf{Pr}(b)]\!]_{\eta_2{}^{(State)}} = p\cdot p_1 + (1-p)\cdot p_2 = (1-p)\cdot p_2 > 0$.
  From $0 = p_1 = [\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}} = \mathbf{Pr}_{\sigma\sim\eta_1{}^{(State)}}[\sigma \models b] = \mathbf{Pr}_{(C,\sigma)\sim\eta_1}[\sigma \models b] = \sum_{C,\sigma}\{\eta_1(C,\sigma) \mid \sigma \models b\}$ we know $\sigma \not\models b$ for all C and σ such that

$\eta_1(C,\sigma) > 0$, thus

$$\begin{aligned}
&(\eta_1 \oplus_p \eta_2)|_b \\
&= \lambda(C,\sigma).\ \tfrac{\chi(\sigma \models b)\cdot(\eta_1 \oplus_p \eta_2)(C,\sigma)}{[\![\mathbf{Pr}(b)]\!]_{(\eta_1 \oplus_p \eta_2)^{(State)}}} \qquad \text{(by Lem. 207)} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(\sigma \models b)\cdot(p\cdot\eta_1(C,\sigma)+(1-p)\cdot\eta_2(C,\sigma))}{(1-p)\cdot p_2} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(\sigma \models b)\cdot(1p)\cdot\eta_2(C,\sigma)}{(1-p)\cdot p_2} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(\sigma \models b)\cdot\eta_2(C,\sigma)}{[\![\mathbf{Pr}(b)]\!]_{\eta_2{}^{(State)}}} \\
&= \eta_2|_b. \qquad \text{(by Lem. 207)}
\end{aligned}$$

– $p_1 = 0 \wedge p_2 = 0$.
By Lem. 12 and Lem. 237 we know $[\![\mathbf{Pr}(b)]\!]_{(\eta_1 \oplus_p \eta_2)^{(State)}} = [\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)} \oplus_p \eta_2{}^{(State)}} = p \cdot [\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}} + (1-p) \cdot [\![\mathbf{Pr}(b)]\!]_{\eta_2{}^{(State)}} = p \cdot p_1 + (1-p) \cdot p_2 = 0$. By Lem. 205 we know $(\eta_1 \oplus_p \eta_2)|_b =$ undefined.

Definition 80. $\mathbf{Nosplit}(\eta)$ if and only if $\mathbf{Nosplit}(C)$ for all $C \in \mathit{supp}(\eta^{(Stmt)})$.

Lemma 239. *For all η and η', if $\mathit{supp}(\eta'^{(Stmt)}) \subseteq \mathit{supp}(\eta^{(Stmt)})$ and $\mathbf{Nosplit}(\eta)$, then $\mathbf{Nosplit}(\eta')$.*

Proof. For all η and η' such that $\mathit{supp}(\eta'^{(Stmt)}) \subseteq \mathit{supp}(\eta^{(Stmt)})$ and $\mathbf{Nosplit}(\eta)$, to prove $\mathbf{Nosplit}(\eta')$, we need to prove $\mathbf{Nosplit}(C)$ for all $C \in \mathit{supp}(\eta'^{(Stmt)})$. For all $C \in \mathit{supp}(\eta'^{(Stmt)})$, from $\mathit{supp}(\eta'^{(Stmt)}) \subseteq \mathit{supp}(\eta^{(Stmt)})$ we know $C \in \mathit{supp}(\eta^{(Stmt)})$. From $\mathbf{Nosplit}(\eta)$ we know $\mathbf{Nosplit}(C)$.

Lemma 240. *For all R, μ, μ', C, if $\mu \overset{R}{\mapsto} \mu'$, then $\delta(C_0) \otimes \mu \overset{R}{\mapsto} \delta(C_0) \otimes \mu'$.*

Proof. For all R, μ, μ', C such that $\mu \overset{R}{\mapsto} \mu'$, there exists θ such that $\mathit{dom}(\theta) = \mathit{supp}(\mu)$, $\mathit{range}(\theta) = \mathit{supp}(\mu')$ and $\theta \subseteq [\![R]\!]$. Let $\psi \overset{\text{def}}{=} \{((C_0,\sigma),(C_0,\sigma')) \mid (\sigma,\sigma') \in \theta\}$, then

$$\begin{aligned}
\mathit{dom}(\psi) &= \{(C,\sigma) \mid \exists C', \sigma'.\ ((C,\sigma),(C',\sigma')) \in \psi\} \\
&= \{(C_0,\sigma) \mid \exists \sigma'.\ (\sigma,\sigma') \in \theta\} \\
&= \{(C_0,\sigma) \mid \sigma \in \mathit{dom}(\theta)\} \\
&= \{(C_0,\sigma) \mid \sigma \in \mathit{supp}(\mu)\} \\
&= \{(C_0,\sigma) \mid \mu(\sigma) > 0\} \\
&= \{(C_0,\sigma) \mid (\delta(C) \otimes \mu)(C,\sigma) > 0\} \\
&= \mathit{supp}(\delta(C_0) \otimes \mu)
\end{aligned}$$

and

$$\begin{aligned}
\mathit{range}(\psi) &= \{(C',\sigma') \mid \exists C, \sigma.\ ((C,\sigma),(C',\sigma')) \in \psi\} \\
&= \{(C_0,\sigma') \mid \exists \sigma.\ (\sigma,\sigma') \in \theta\} \\
&= \{(C_0,\sigma') \mid \sigma' \in \mathit{range}(\theta)\} \\
&= \{(C_0,\sigma') \mid \sigma' \in \mathit{supp}(\mu')\} \\
&= \{(C_0,\sigma') \mid \mu'(\sigma') > 0\} \\
&= \{(C_0,\sigma') \mid (\delta(C) \otimes \mu')(C,\sigma') > 0\} \\
&= \mathit{supp}(\delta(C_0) \otimes \mu').
\end{aligned}$$

For all $((C,\sigma),(C',\sigma')) \in \psi$, we have $C' = C = C_0$ and $(\sigma,\sigma') \in \theta$. From $\theta \subseteq [\![R]\!]$ we know $(\sigma,\sigma') \in [\![R]\!]$, thus $(\sigma,\sigma') \models R$. Therefore, $\delta(C_0) \otimes \mu \overset{R}{\mapsto} \delta(C_0) \otimes \mu'$.

Lemma 241. *For all R, I, G, Q, η, if $(\eta, R, I) \Longmapsto^n_{NST} (G, Q)$ for all n, then the following are true:*

- *if $\eta^{(Stmt)}(\textbf{\textit{skip}}) > 0$, then ${\eta|_{\textbf{\textit{skip}}}}^{(State)} \models Q$.*
- *$\eta^{(State)} \models I$.*
- *for all η', if $\eta \underset{I}{\overset{R}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longmapsto^n_{NST} (G, Q)$ for all n.*
- *for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, ${\eta'}^{(State)} \models I$ and $(\eta', R, I) \Longmapsto^n_{NST} (G, Q)$ for all n.*

Proof. For all R, I, G, Q, η such that $(\eta, R, I) \Longmapsto^n_{\text{NST}} (G, Q)$ for all n, we need to prove

- if $\eta^{(Stmt)}(\textbf{skip}) > 0$, then ${\eta|_{\textbf{skip}}}^{(State)} \models Q$.
  From $(\eta, R, I) \Longmapsto^n_{\text{NST}} (G, Q)$ for all n we know $(\eta, R, I) \Longmapsto^1_{\text{NST}} (G, Q)$. From $\eta^{(Stmt)}(\textbf{skip}) > 0$ we know ${\eta|_{\textbf{skip}}}^{(State)} \models Q$.
- $\eta^{(State)} \models I$.
  From $(\eta, R, I) \Longmapsto^n_{\text{NST}} (G, Q)$ for all n we know $(\eta, R, I) \Longmapsto^1_{\text{NST}} (G, Q)$, thus $\eta^{(State)} \models I$.
- for all η', if $\eta \underset{I}{\overset{R}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longmapsto^n_{\text{NST}} (G, Q)$ for all n.

  For all η' and n such that $\eta \underset{I}{\overset{R}{\rightarrowtail}} \eta'$, from $(\eta, R, I) \Longmapsto^{n+1}_{\text{NST}} (G, Q)$ we know $(\eta', R, I) \Longmapsto^n_{\text{NST}} (G, Q)$.
- for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, ${\eta'}^{(State)} \models I$ and $(\eta', R, I) \Longmapsto^n_{\text{NST}} (G, Q)$ for all n. For all θ, η', n such that $\eta \hookrightarrow (\theta, \eta')$, from $(\eta, R, I) \Longmapsto^{n+1}_{\text{NST}} (G, Q)$ we know $\theta \subseteq [\![G]\!]$, ${\eta'}^{(State)} \models I$ and $(\eta', R, I) \Longmapsto^n_{\text{NST}} (G, Q)$.

Lemma 242. *For all ρ and μ, if $[\![\textbf{\textit{Pr}}(b)]\!]_\mu > 0$, then $(\rho \otimes \mu)|_b = \rho \otimes \mu|_b$.*

Proof. For all ρ and μ such that $[\![\mathbf{Pr}(b)]\!]_\mu > 0$, by Lem. 19 we know $(\rho \otimes \mu)^{(State)} = \mu$, thus $[\![\mathbf{Pr}(b)]\!]_{(\rho\otimes\mu)} = [\![\mathbf{Pr}(b)]\!]_\mu > 0$. By Lem. 207 we have

$$
\begin{aligned}
& (\rho \otimes \mu)|_b \\
= {} & \lambda(C,\sigma).\ \tfrac{\chi(\sigma \models b) \cdot (\rho\otimes\mu)(C,\sigma)}{[\![\mathbf{Pr}(b)]\!]_\mu} \\
= {} & \lambda(C,\sigma).\ \tfrac{\chi(\sigma \models b) \cdot \rho(C) \cdot \mu(\sigma)}{[\![\mathbf{Pr}(b)]\!]_\mu} \\
= {} & \lambda(C,\sigma).\ \rho(C) \cdot \tfrac{\chi(\sigma \models b) \cdot \mu(\sigma)}{\mathbf{Pr}_{\sigma'\sim\mu}[\sigma' \models b]} \\
= {} & \lambda(C,\sigma).\ \rho(C) \cdot \mu|_b(\sigma) \\
= {} & \rho \otimes \mu|_b.
\end{aligned}
$$

Lemma 243. *For all η_1, η_2, p, if $0 < p < 1$, then $\mathit{nextsplit}(\eta_1 \oplus_p \eta_2) = \mathit{nextsplit}(\eta_1) \cup \mathit{nextsplit}(\eta_2)$.*

Proof. For all η_1, η_2, p such that $0 < p < 1$, by Lem. 275 we have $supp(\eta_1 \oplus_p \eta_2) = supp(\eta_1) \cup supp(\eta_2)$, thus

$$
\begin{aligned}
& nextsplit(\eta_1 \oplus_p \eta_2) \\
&= \{nextsplit(C) \mid \exists \sigma.\ (C,\sigma) \in supp(\eta_1 \oplus_p \eta_2)\} \\
&= \{nextsplit(C) \mid \exists \sigma.\ (C,\sigma) \in supp(\eta_1) \cup supp(\eta_2)\} \\
&= \{nextsplit(C) \mid \exists \sigma.\ (C,\sigma) \in supp(\eta_1) \vee (C,\sigma) \in supp(\eta_2)\} \\
&= \{nextsplit(C) \mid (\exists \sigma.\ (C,\sigma) \in supp(\eta_1)) \vee (\exists \sigma.\ (C,\sigma) \in supp(\eta_2))\} \\
&= \{nextsplit(C) \mid \exists \sigma.\ (C,\sigma) \in supp(\eta_1)\} \cup \{nextsplit(C) \mid \exists \sigma.\ (C,\sigma) \in supp(\eta_2)\} \\
&= nextsplit(\eta_1) \cup nextsplit(\eta_2).
\end{aligned}
$$

Lemma 244. *For all η, if* $\mathbf{Nosplit}(\eta)$*, then* $nextsplit(\eta) = \{\boldsymbol{split}(true)\}$.

Proof. For all η such that $\mathbf{Nosplit}(\eta)$, we know $\mathbf{Nosplit}(C)$ for all $C \in supp(\eta^{(Stmt)})$, thus

$$
\begin{aligned}
& nextsplit(\eta) \\
&= \{(nextsplit(C)) \mid \exists \sigma.\ (C,\sigma) \in supp(\eta)\} \\
&= \{(nextsplit(C)) \mid \exists \sigma.\ \eta(C,\sigma) > 0\} \\
&= \{(nextsplit(C) \mid \textstyle\sum_\sigma \eta(C,\sigma) > 0\} \\
&= \{(nextsplit(C) \mid \eta^{(Stmt)}(C) > 0\} \\
&= \{(nextsplit(C) \mid C \in supp(\eta^{(Stmt)})\} \\
&= \{(nextsplit(C) \mid C \in supp(\eta^{(Stmt)}) \wedge \mathbf{Nosplit}(C)\} \\
&= \{(nextsplit(C) \mid C \in supp(\eta^{(Stmt)}) \wedge \mathbf{Nosplit}(C) \wedge nextsplit(C) = \mathbf{split}(\mathrm{true})\} \qquad \text{(by Lem. 45)} \\
&= \{\mathbf{split}(\mathrm{true})\}.
\end{aligned}
$$

Lemma 245. *For all $\eta_1, \eta_2, p, \theta, \eta'$, if $0 < p < 1$ and $(\eta_1 \oplus_p \eta_2) \rightsquigarrow (\theta, \eta')$, then there exists $\theta_1, \theta_2, \eta_1', \eta_2'$ such that $\theta = \theta_1 \cup \theta_2$, $\eta' = \eta_1' \oplus_p \eta_2'$, $\eta_1 \rightsquigarrow (\theta_1, \eta_1')$ and $\eta_2 \rightsquigarrow (\theta_2, \eta_2')$.*

Proof. For all $\eta_1, \eta_2, p, \theta, \eta'$ such that $0 < p < 1$ and $(\eta_1 \oplus_p \eta_2) \rightsquigarrow (\theta, \eta')$, let $\eta_1' \stackrel{\text{def}}{=} \lambda(C', \sigma').$
$\sum_{C,\sigma}\{\eta_1(C,\sigma) \cdot p \mid (C,\sigma) \xrightarrow{p} (C',\sigma')\}$, $\theta_1 \stackrel{\text{def}}{=} \{(\sigma,\sigma') \mid \exists C, C'.\ \eta_1(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}$, $\eta_2' \stackrel{\text{def}}{=} \lambda(C',\sigma').\sum_{C,\sigma}\{\eta_2(C,\sigma)\cdot p \mid (C,\sigma) \xrightarrow{p} (C',\sigma')\}$ and $\theta_2 \stackrel{\text{def}}{=} \{(\sigma,\sigma') \mid \exists C, C'.\ \eta_2(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}$, we have $\eta_1 \rightsquigarrow (\theta_1, \eta_1')$ and $\eta_2 \rightsquigarrow (\theta_2, \eta_2')$. From $0 < p < 1$ by Lem. 246 we know $(\eta_1 \oplus_p \eta_2) \rightsquigarrow (\theta_1 \cup \theta_2, \eta_1' \oplus_p \eta_2')$. From $(\eta_1 \oplus_p \eta_2) \rightsquigarrow (\theta, \eta')$ by Lem. 193 we have $\theta = \theta_1 \cup \theta_2$ and $\eta' = \eta_1' \oplus_p \eta_2'$.

Lemma 246. *For all $\eta_1, \eta_2, p, \theta_1, \theta_2, \eta_1', \eta_2'$, if $0 < p < 1$, $\eta_1 \rightsquigarrow (\theta_1, \eta_1')$ and $\eta_2 \rightsquigarrow (\theta_2, \eta_2')$, then $(\eta_1 \oplus_p \eta_2) \rightsquigarrow (\theta_1 \cup \theta_2, \eta_1' \oplus_p \eta_2')$.*

Proof. For all $\eta_1, \eta_2, p, \theta_1, \theta_2, \eta_1', \eta_2'$ such that $0 < p < 1$, $\eta_1 \rightsquigarrow (\theta_1, \eta_1')$ and $\eta_2 \rightsquigarrow (\theta_2, \eta_2')$, we have $\eta_1' = \lambda(C',\sigma').\sum_{C,\sigma}\{\eta_1(C,\sigma)\cdot p \mid (C,\sigma) \xrightarrow{p} (C',\sigma')\}$, $\theta_1 = \{(\sigma,\sigma') \mid \exists C, C'.\ \eta_1(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}$, $\eta_2' =$

$\lambda(C',\sigma').\sum_{C,\sigma}\{\eta_2(C,\sigma)\cdot p \mid (C,\sigma) \xrightarrow{p} (C',\sigma')\}$ and $\theta_2 = \{(\sigma,\sigma') \mid \exists C, C'.\ \eta_2(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}$, thus

$$\begin{aligned}
\eta_1' \oplus_p \eta_2' &= \lambda(C',\sigma').\ p\cdot\eta_1'(C',\sigma') + (1-p)\cdot\eta_2'(C',\sigma') \\
&= \lambda(C',\sigma').\ p\cdot\textstyle\sum_{C,\sigma}\{\eta_1(C,\sigma)\cdot p' \mid (C,\sigma) \xrightarrow{p'} (C',\sigma')\} + \\
&\qquad (1-p)\cdot\textstyle\sum_{C,\sigma}\{\eta_2(C,\sigma)\cdot p' \mid (C,\sigma) \xrightarrow{p'} (C',\sigma')\} \\
&= \lambda(C',\sigma').\textstyle\sum_{C,\sigma}\{p\cdot\eta_1(C,\sigma) + (1-p)\cdot\eta_2(C,\sigma)\cdot p' \mid (C,\sigma) \xrightarrow{p'} (C',\sigma')\} \\
&= \lambda(C',\sigma').\textstyle\sum_{C,\sigma}\{(\eta_1 \oplus_p \eta_2)(C,\sigma)\cdot p' \mid (C,\sigma) \xrightarrow{p'} (C',\sigma')\}.
\end{aligned}$$

From $0 < p < 1$ by Lem. 275 we know $supp(\eta_1 \oplus_p \eta_2) = supp(\eta_1) \cup supp(\eta_2)$, thus

$$\begin{aligned}
&\theta_1 \cup \theta_2 \\
&= \{(\sigma,\sigma') \mid \exists C, C'.\ \eta_1(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p'} (C',\sigma') \wedge p' > 0\} \cup \\
&\quad \{(\sigma,\sigma') \mid \exists C, C'.\ \eta_2(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p'} (C',\sigma') \wedge p' > 0\} \\
&= \{(\sigma,\sigma') \mid \exists C, C'.\ ((C,\sigma) \in supp(\eta_1) \vee (C,\sigma) \in supp(\eta_2)) \wedge (C,\sigma) \xrightarrow{p'} (C',\sigma') \wedge p' > 0\} \\
&= \{(\sigma,\sigma') \mid \exists C, C'.\ (C,\sigma) \in supp(\eta_1) \cup supp(\eta_2) \wedge (C,\sigma) \xrightarrow{p'} (C',\sigma') \wedge p' > 0\} \\
&= \{(\sigma,\sigma') \mid \exists C, C'.\ (C,\sigma) \in supp(\eta_1 \oplus_p \eta_2) \wedge (C,\sigma) \xrightarrow{p'} (C',\sigma') \wedge p' > 0\}.
\end{aligned}$$

Therefore, $(\eta_1 \oplus_p \eta_2) \rightsquigarrow (\theta_1 \cup \theta_2, \eta_1' \oplus_p \eta_2')$.

Lemma 247. *For all η, if $0 < \eta^{(Stmt)}(\boldsymbol{skip}) < 1$, then there exists η_1 and η_2 such that $\eta = \eta_1 \oplus_{\eta^{(Stmt)}(\boldsymbol{skip})} \eta_2$, ${\eta_1}^{(Stmt)}(\boldsymbol{skip}) = 1$ and ${\eta_2}^{(Stmt)}(\boldsymbol{skip}) = 0$.*

Proof. For all η such that $0 < \eta^{(Stmt)}(\mathbf{skip}) < 1$, let $\eta_1 \stackrel{\text{def}}{=} \lambda(C,\sigma).\frac{\chi(C=\mathbf{skip})\cdot\eta(C,\sigma)}{\eta^{(Stmt)}(\mathbf{skip})}$ and $\eta_2 \stackrel{\text{def}}{=} \lambda(C,\sigma).\frac{\chi(C\neq\mathbf{skip})\cdot\eta(C,\sigma)}{1-\eta^{(Stmt)}(\mathbf{skip})}$, then

$$\begin{aligned}
&\eta_1 \oplus_{\eta^{(Stmt)}(\mathbf{skip})} \eta_2 \\
&= \lambda(C,\sigma).\ \eta^{(Stmt)}(\mathbf{skip})\cdot\eta_1(C,\sigma) + (1-\eta^{(Stmt)}(\mathbf{skip}))\cdot\eta_2(C,\sigma) \\
&= \lambda(C,\sigma).\ \eta^{(Stmt)}(\mathbf{skip})\cdot\tfrac{\chi(C=\mathbf{skip})\cdot\eta(C,\sigma)}{\eta^{(Stmt)}(\mathbf{skip})} + (1-\eta^{(Stmt)}(\mathbf{skip}))\cdot\tfrac{\chi(C\neq\mathbf{skip})\cdot\eta(C,\sigma)}{1-\eta^{(Stmt)}(\mathbf{skip})} \\
&= \lambda(C,\sigma).\ \chi(C=\mathbf{skip})\cdot\eta(C,\sigma) + \chi(C\neq\mathbf{skip})\cdot\eta(C,\sigma) \\
&= \lambda(C,\sigma).\ \eta(C,\sigma) \\
&= \eta,
\end{aligned}$$

$$\begin{aligned}
{\eta_1}^{(Stmt)}(\mathbf{skip}) &= \textstyle\sum_\sigma \eta_1(\mathbf{skip},\sigma) \\
&= \textstyle\sum_\sigma \frac{\chi(\mathbf{skip}=\mathbf{skip})\cdot\eta(\mathbf{skip},\sigma)}{\eta^{(Stmt)}(\mathbf{skip})} \\
&= \frac{\sum_\sigma \eta(\mathbf{skip},\sigma)}{\eta^{(Stmt)}(\mathbf{skip})} \\
&= 1,
\end{aligned}$$

and

$$\begin{aligned}
{\eta_2}^{(Stmt)}(\mathbf{skip}) &= \textstyle\sum_\sigma \eta_2(\mathbf{skip},\sigma) \\
&= \textstyle\sum_\sigma \frac{\chi(\mathbf{skip}\neq\mathbf{skip})\cdot\eta(\mathbf{skip},\sigma)}{1-\eta^{(Stmt)}(\mathbf{skip})} \\
&= 0.
\end{aligned}$$

Lemma 248. *For all* η_1, η_2, p, C_2, $(\eta_1 \oplus_p \eta_2); C_2 = \eta_1; C_2 \oplus_p \eta_2; C_2$.

Proof. For all η_1, η_2, p, C_2,

$$
\begin{aligned}
&(\eta_1 \oplus_p \eta_2); C_2 \\
&= \lambda(C,\sigma).\ \begin{cases} (\eta_1 \oplus_p \eta_2)(C_1,\sigma), & \text{if } C = C_1; C_2 \\ 0, & \text{otherwise} \end{cases} \\
&= \lambda(C,\sigma).\ \begin{cases} p \cdot \eta_1(C_1,\sigma) + (1-p) \cdot \eta_2(C_1,\sigma), & \text{if } C = C_1; C_2 \\ 0, & \text{otherwise} \end{cases} \\
&= \lambda(C,\sigma).\ p \cdot \begin{cases} \eta_1(C_1,\sigma), & \text{if } C = C_1; C_2 \\ 0, & \text{otherwise} \end{cases} + (1-p) \cdot \begin{cases} \eta_2(C_1,\sigma), & \text{if } C = C_1; C_2 \\ 0, & \text{otherwise} \end{cases} \\
&= \lambda(C,\sigma).\ p \cdot (\eta_1; C_2)(C,\sigma) + (1-p) \cdot (\eta_2; C_2)(C,\sigma) \\
&= \eta_1; C_2 \oplus_p \eta_2; C_2.
\end{aligned}
$$

We use $VS \in \mathcal{P}(PVar)$ to denote the set of program variables.

Definition 81. $\sigma|_{VS} \stackrel{\text{def}}{=} \lambda x \in VS.\ \sigma(x)$.

Definition 82. $\mu|_{VS} \stackrel{\text{def}}{=} \lambda \hat{\sigma} \in VS \to \mathbb{R}.\ \sum_{\sigma} \{\mu(\sigma) \mid \sigma|_{VS} = \hat{\sigma}\}$.

Lemma 249. *For all* η, θ, η', VS, *if* $\eta \rightsquigarrow (\theta, \eta')$ *and* $\sigma'(x) = \sigma(x)$ *for all* x, σ, σ' *such that* $x \in VS$ *and* $(\sigma, \sigma') \in \theta$, *then* ${\eta'}^{(State)}|_{VS} = \eta^{(State)}|_{VS}$.

Proof. For all η, θ, η', VS such that $\eta \rightsquigarrow (\theta, \eta')$ and $\sigma'(x) = \sigma(x)$ for all x, σ, σ' such that $x \in VS$ and $(\sigma, \sigma') \in \theta$, from $\eta \rightsquigarrow (\theta, \eta')$ we know $\eta' = \lambda(C', \sigma').\ \sum_{C,\sigma} \{\eta(C,\sigma) \cdot p \mid (C,\sigma) \xrightarrow{p} (C', \sigma')\}$ and $\theta = \{(\sigma, \sigma') \mid \eta(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0\}$. From $\sigma'(x) = \sigma(x)$ for all x, σ, σ' such that $x \in VS$ and $(\sigma, \sigma') \in \theta$ we have for all $(\sigma, \sigma') \in \theta$, $\sigma'|_{VS} = \lambda x \in VS.\ \sigma'(x) = \lambda x \in VS.\ \sigma(x) = \sigma|_{VS}$, thus

$$
\begin{aligned}
&{\eta'}^{(State)}|_{VS} \\
&= \lambda \hat{\sigma} \in VS \to \mathbb{R}.\ \textstyle\sum_{\sigma'} \{{\eta'}^{(State)}(\sigma') \mid \sigma'|_{VS} = \hat{\sigma}\} \\
&= \lambda \hat{\sigma} \in VS \to \mathbb{R}.\ \textstyle\sum_{\sigma', C'} \{\eta'(C', \sigma') \mid \sigma'|_{VS} = \hat{\sigma}\} \\
&= \lambda \hat{\sigma} \in VS \to \mathbb{R}.\ \textstyle\sum_{\sigma', C', C, \sigma} \{\eta(C,\sigma) \cdot p \mid (C,\sigma) \xrightarrow{p} (C', \sigma') \wedge \sigma'|_{VS} = \hat{\sigma}\} \\
&= \lambda \hat{\sigma} \in VS \to \mathbb{R}.\ \textstyle\sum_{\sigma', C', C, \sigma} \{\eta(C,\sigma) \cdot p \mid \eta(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C', \sigma') \wedge p > 0 \wedge \sigma'|_{VS} = \hat{\sigma}\} \\
&= \lambda \hat{\sigma} \in VS \to \mathbb{R}.\ \textstyle\sum_{\sigma', C', C, \sigma} \{\eta(C,\sigma) \cdot p \mid \eta(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C', \sigma') \wedge \\
&\qquad p > 0 \wedge (\sigma, \sigma') \in \theta \wedge \sigma'|_{VS} = \hat{\sigma}\} \\
&= \lambda \hat{\sigma} \in VS \to \mathbb{R}.\ \textstyle\sum_{\sigma', C', C, \sigma} \{\eta(C,\sigma) \cdot p \mid \eta(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C', \sigma') \wedge \\
&\qquad p > 0 \wedge (\sigma, \sigma') \in \theta \wedge \sigma|_{VS} = \hat{\sigma}\} \\
&= \lambda \hat{\sigma} \in VS \to \mathbb{R}.\ \textstyle\sum_{\sigma', C', C, \sigma} \{\eta(C,\sigma) \cdot p \mid (C,\sigma) \xrightarrow{p} (C', \sigma') \wedge \sigma|_{VS} = \hat{\sigma}\} \\
&= \lambda \hat{\sigma} \in VS \to \mathbb{R}.\ \textstyle\sum_{C,\sigma} \{\eta(C,\sigma) \cdot \sum_{C', \sigma'} \{p \mid (C,\sigma) \xrightarrow{p} (C', \sigma')\} \mid \sigma|_{VS} = \hat{\sigma}\} \\
&= \lambda \hat{\sigma} \in VS \to \mathbb{R}.\ \textstyle\sum_{C,\sigma} \{\eta(C,\sigma) \mid \sigma|_{VS} = \hat{\sigma}\} \\
&= \lambda \hat{\sigma} \in VS \to \mathbb{R}.\ \textstyle\sum_{\sigma} \{\eta^{(State)}(\sigma) \mid \sigma|_{VS} = \hat{\sigma}\} \\
&= \eta^{(State)}|_{VS}.
\end{aligned}
$$

Lemma 250. *For all* μ_1, μ_2, p, VS, $(\mu_1 \oplus_p \mu_2)|_{VS} = \mu_1|_{VS} \oplus_p \mu_2|_{VS}$.

Proof. For all μ_1, μ_2, p, VS, we have

$$
\begin{aligned}
&(\mu_1 \oplus_p \mu_2)|_{VS} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}.\, \textstyle\sum_\sigma \{(\mu_1 \oplus_p \mu_2)(\sigma) \mid \sigma|_{VS} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}.\, \textstyle\sum_\sigma \{p \cdot \mu_1(\sigma) + (1-p) \cdot \mu_2(\sigma) \mid \sigma|_{VS} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}.\, p \cdot \textstyle\sum_\sigma \{\mu_1(\sigma) \mid \sigma|_{VS} = \hat{\sigma}\} + (1-p) \cdot \sum_\sigma \{\mu_2(\sigma) \mid \sigma|_{VS} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}.\, p \cdot \mu_1|_{VS}(\hat{\sigma}) + (1-p) \cdot \mu_2|_{VS}(\hat{\sigma}) \\
&= \mu_1|_{VS} \oplus_p \mu_2|_{VS}.
\end{aligned}
$$

Lemma 251. *For all* $VS, VS', \sigma_1, \sigma_2$, *if* $VS' \subseteq VS$ *and* $\sigma_1|_{VS} = \sigma_2|_{VS}$, *then* $\sigma_1|_{VS'} = \sigma_2|_{VS'}$.

Proof. For all $VS, VS', \sigma_1, \sigma_2$ such that $VS' \subseteq VS$ and $\sigma_1|_{VS} = \sigma_2|_{VS}$, to prove $\sigma_1|_{VS'} = \sigma_2|_{VS'}$, we need to prove $\sigma_1|_{VS'}(x) = \sigma_2|_{VS'}(x)$ for all $x \in VS'$. For all $x \in VS'$, from $VS' \subseteq VS$ we know $x \in VS$, thus $\sigma_1|_{VS'}(x) = \sigma_1(VS') = \sigma_1|_{VS}(x) = \sigma_2|_{VS}(x) = \sigma_2|_{VS'}$.

Lemma 252. *For all* e, σ_1, σ_2, *if* $\sigma_1|_{fv(e)} = \sigma_2|_{fv(e)}$, *then* $[\![e]\!]_{\sigma_1} = [\![e]\!]_{\sigma_2}$.

Proof. by induction on e.

- case n.
  For all σ_1, σ_2 such that $\sigma_1|_{fv(n)} = \sigma_2|_{fv(n)}$, we have $[\![n]\!]_{\sigma_1} = n = [\![n]\!]_{\sigma_2}$.
- case x.
  For all σ_1, σ_2 such that $\sigma_1|_{fv(x)} = \sigma_2|_{fv(x)}$, thus $\sigma_1|_{fv(x)}(x) = \sigma_2|_{fv(x)}(x)$. From $x \in fv(x) = \{x\}$ we know $\sigma_1|_{fv(x)}(x) = \sigma_1(x)$ and $\sigma_2|_{fv(x)}(x) = \sigma_2(x)$, thus $\sigma_1(x) = \sigma_2(x)$. Therefore $[\![x]\!]_{\sigma_1} = \sigma_1(x) = \sigma_2(x) = [\![x]\!]_{\sigma_2}$.
- case $e_1 + e_2$.
  IH1: for all σ_1, σ_2, if $\sigma_1|_{fv(e_1)} = \sigma_2|_{fv(e_1)}$, then $[\![e_1]\!]_{\sigma_1} = [\![e_1]\!]_{\sigma_2}$.
  IH2: for all σ_1, σ_2, if $\sigma_1|_{fv(e_2)} = \sigma_2|_{fv(e_2)}$, then $[\![e_2]\!]_{\sigma_1} = [\![e_2]\!]_{\sigma_2}$.
  For all σ_1, σ_2 such that $\sigma_1|_{fv(e_1+e_2)} = \sigma_2|_{fv(e_1+e_2)}$, from $fv(e_1 + e_2) = fv(e_1) \cup fv(e_2)$ we know $fv(e_1) \subseteq fv(e_1 + e_2)$. From $\sigma_1|_{fv(e_1+e_2)} = \sigma_2|_{fv(e_1<e_2)}$ by Lem. 251 we know $\sigma_1|_{fv(e_1)} = \sigma_2|_{fv(e_1)}$. By IH1 we have $[\![e_1]\!]_{\sigma_1} = [\![e_1]\!]_{\sigma_2}$. Similarly we can prove $[\![e_2]\!]_{\sigma_1} = [\![e_2]\!]_{\sigma_2}$. Therefore $[\![e_1 + e_2]\!]_{\sigma_1} = [\![e_1]\!]_{\sigma_1} + [\![e_2]\!]_{\sigma_1} = [\![e_1]\!]_{\sigma_2} + [\![e_2]\!]_{\sigma_2} = [\![e_1 < e_2]\!]_{\sigma_2}$.
- case $e_1 - e_2$.
  Similar to the case $e_1 + e_2$.
- case $e_1 * e_2$.
  Similar to the case $e_1 + e_2$.

Lemma 253. *For all* b, σ_1, σ_2, *if* $\sigma_1|_{fv(b)} = \sigma_2|_{fv(b)}$, *then* $[\![b]\!]_{\sigma_1} = [\![b]\!]_{\sigma_2}$.

Proof. by induction on b.

- case true.
  For all σ_1, σ_2 such that $\sigma_1|_{fv(\mathrm{true})} = \sigma_2|_{fv(\mathrm{true})}$, we have $[\![\mathrm{true}]\!]_{\sigma_1} = \mathrm{tt} = [\![\mathrm{true}]\!]_{\sigma_2}$.

- case false.
  For all σ_1, σ_2 such that $\sigma_1|_{fv(\text{false})} = \sigma_2|_{fv(\text{false})}$, we have $[\![\text{false}]\!]_{\sigma_1} = \text{ff} = [\![\text{false}]\!]_{\sigma_2}$.
- case $e_1 < e_2$.
  For all σ_1, σ_2 such that $\sigma_1|_{fv(e_1<e_2)} = \sigma_2|_{fv(e_1<e_2)}$, from $fv(e_1 < e_2) = fv(e_1) \cup fv(e_2)$ we know $fv(e_1) \subseteq fv(e_1 < e_2)$. From $\sigma_1|_{fv(e_1<e_2)} = \sigma_2|_{fv(e_1<e_2)}$ by Lem. 251 we know $\sigma_1|_{fv(e_1)} = \sigma_2|_{fv(e_1)}$. By Lem. 252 we have $[\![e_1]\!]_{\sigma_1} = [\![e_1]\!]_{\sigma_2}$. Similarly we can prove $[\![e_2]\!]_{\sigma_1} = [\![e_2]\!]_{\sigma_2}$. Therefore $[\![e_1 < e_2]\!]_{\sigma_1} = \begin{cases} \text{tt}, & \text{if } [\![e_1]\!]_{\sigma_1} < [\![e_2]\!]_{\sigma_1} \\ \text{ff}, & \text{otherwise} \end{cases} = \begin{cases} \text{tt}, & \text{if } [\![e_1]\!]_{\sigma_2} < [\![e_2]\!]_{\sigma_2} \\ \text{ff}, & \text{otherwise} \end{cases} = [\![e_1 < e_2]\!]_{\sigma_2}$.
- case $e_1 = e_2$.
  Similar to the case $e_1 < e_2$.
- case $e_1 \leq e_2$.
  Similar to the case $e_1 < e_2$.
- case $\neg b$.
  IH: for all σ_1, σ_2, if $\sigma_1|_{fv(b)} = \sigma_2|_{fv(b)}$, then $[\![b]\!]_{\sigma_1} = [\![b]\!]_{\sigma_2}$.
  For all σ_1, σ_2 such that $\sigma_1|_{fv(\neg b)} = \sigma_2|_{fv(\neg b)}$, from $fv(\neg b) = fv(b)$ we know $\sigma_1|_{fv(b)} = \sigma_2|_{fv(b)}$. By IH we have $[\![b]\!]_{\sigma_1} = [\![b]\!]_{\sigma_2}$. Therefore $[\![\neg b]\!]_{\sigma_1} = \begin{cases} \text{ff}, & \text{if } [\![b]\!]_{\sigma_1} = \text{tt} \\ \text{tt}, & \text{otherwise} \end{cases} = \begin{cases} \text{ff}, & \text{if } [\![b]\!]_{\sigma_2} = \text{tt} \\ \text{tt}, & \text{otherwise} \end{cases} = [\![\neg b]\!]_{\sigma_2}$.
- case $b_1 \wedge b_2$.
  IH1: for all σ_1, σ_2, if $\sigma_1|_{fv(b_1)} = \sigma_2|_{fv(b_1)}$, then $[\![b_1]\!]_{\sigma_1} = [\![b_1]\!]_{\sigma_2}$.
  IH2: for all σ_1, σ_2, if $\sigma_1|_{fv(b_2)} = \sigma_2|_{fv(b_2)}$, then $[\![b_2]\!]_{\sigma_1} = [\![b_2]\!]_{\sigma_2}$.
  For all σ_1, σ_2 such that $\sigma_1|_{fv(b_1 \wedge b_2)} = \sigma_2|_{fv(b_1 \wedge b_2)}$, from $fv(b_1 \wedge b_2) = fv(b_1) \cup fv(b_2)$ we know $fv(b_1) \subseteq fv(b_1 \wedge b_2)$. From $\sigma_1|_{fv(b_1 \wedge b_2)} = \sigma_2|_{fv(b_1 \wedge b_2)}$ by Lem. 251 we have $\sigma_1|_{fv(b_1)} = \sigma_2|_{fv(b_1)}$. By IH1 we have $[\![b_1]\!]_{\sigma_1} = [\![b_1]\!]_{\sigma_2}$. Similarly we can prove $[\![b_2]\!]_{\sigma_1} = [\![b_2]\!]_{\sigma_2}$. Therefore $[\![b_1 \wedge b_2]\!]_{\sigma_1} = \begin{cases} \text{tt}, & \text{if } [\![b_1]\!]_{\sigma_1} = \text{tt} \text{ and } [\![b_2]\!]_{\sigma_1} = \text{tt} \\ \text{ff}, & \text{otherwise} \end{cases}$
  $= \begin{cases} \text{tt}, & \text{if } [\![b_1]\!]_{\sigma_2} = \text{tt} \text{ and } [\![b_2]\!]_{\sigma_2} = \text{tt} \\ \text{ff}, & \text{otherwise} \end{cases} = [\![b_1 \wedge b_2]\!]_{\sigma_2}$.
- case $b_1 \vee b_2$.
  Similar to the case $b_1 \wedge b_2$.

Lemma 254. *For all $VS, X, r, \sigma_1, \sigma_2$, if $\sigma_1|_{VS-\{X\}} = \sigma_2|_{VS-\{X\}}$, then $\sigma_1\{X \rightsquigarrow r\}|_{VS} = \sigma_2\{X \rightsquigarrow r\}|_{VS}$.*

Proof. For all $VS, X, r, \sigma_1, \sigma_2$ such that $\sigma_1|_{VS-\{X\}} = \sigma_2|_{VS-\{X\}}$, to prove $\sigma_1\{X \rightsquigarrow r\}|_{VS} = \sigma_2\{X \rightsquigarrow r\}|_{VS}$, we need to prove $\sigma_1\{X \rightsquigarrow r\}|_{VS}(x) = \sigma_2\{X \rightsquigarrow r\}|_{VS}(x)$ for all $x \in VS$. For all $x \in VS$, we need to prove $\sigma_1\{X \rightsquigarrow r\}(x) = \sigma_2\{X \rightsquigarrow r\}(x)$. If $x = X$, then $\sigma_1\{X \rightsquigarrow r\}(x) = r = \sigma_2\{X \rightsquigarrow r\}(x)$. If $x \neq X$, then $x \in VS - \{X\}$, thus $\sigma_1\{X \rightsquigarrow r\}(x) = \sigma_1(x) = \sigma_1|_{VS-\{X\}}(x) = \sigma_2|_{VS-\{X\}}(x) = \sigma_2(x) = \sigma_2\{X \rightsquigarrow r\}(x)$.

Lemma 255. *For all $\mathbf{q}, \sigma_1, \sigma_2$, if $\sigma_1|_{fv(\mathbf{q})} = \sigma_2|_{fv(\mathbf{q})}$, then $\sigma_1 \models \mathbf{q}$ if and only if $\sigma_2 \models \mathbf{q}$.*

Proof. by induction on $\mathbf{q}$. We only prove one direction (if $\sigma_1 \models \mathbf{q}$ then $\sigma_2 \models \mathbf{q}$) in each case. The other direction is similar.

- case b.
  For all σ_1, σ_2 such that $\sigma_1|_{fv(b)} = \sigma_2|_{fv(b)}$ and $\sigma_1 \models b$, we know $[\![b]\!]_{\sigma_1} = \mathrm{tt}$. From $\sigma_1|_{fv(b)} = \sigma_2|_{fv(b)}$ by Lem. 253 we know $[\![b]\!]_{\sigma_2} = [\![b]\!]_{\sigma_1} = \mathrm{tt}$, thus $\sigma_2 \models b$.
- IH: for all σ_1, σ_2, if $\sigma_1|_{fv(\mathbf{q})} = \sigma_2|_{fv(\mathbf{q})}$, then $\sigma_1 \models \mathbf{q}$ iff $\sigma_2 \models \mathbf{q}$.
  For all σ_1, σ_2 such that $\sigma_1|_{fv(\neg\mathbf{q})} = \sigma_2|_{fv(\neg\mathbf{q})}$ and $\sigma_1 \models \neg\mathbf{q}$, from $fv(\neg\mathbf{q}) = fv(\mathbf{q})$ we know $\sigma_1|_{fv(\mathbf{q})} = \sigma_2|_{fv(\mathbf{q})}$. From $\sigma_1 \models \neg\mathbf{q}$ we know $\sigma_1 \models \mathbf{q}$ does not hold. By IH we know $\sigma_2 \models \mathbf{q}$ does not hold, thus $\sigma_2 \models \neg\mathbf{q}$.
- case $\mathbf{q}_1 \wedge \mathbf{q}_2$.
  IH1: for all σ_1, σ_2, if $\sigma_1|_{fv(\mathbf{q}_1)} = \sigma_2|_{fv(\mathbf{q}_1)}$, then $\sigma_1 \models \mathbf{q}_1$ iff $\sigma_2 \models \mathbf{q}_1$.
  IH2: for all σ_1, σ_2, if $\sigma_1|_{fv(\mathbf{q}_2)} = \sigma_2|_{fv(\mathbf{q}_2)}$, then $\sigma_1 \models \mathbf{q}_2$ iff $\sigma_2 \models \mathbf{q}_2$.
  For all σ_1, σ_2 such that $\sigma_1|_{fv(\mathbf{q}_1\wedge\mathbf{q}_2)} = \sigma_2|_{fv(\mathbf{q}_1\wedge\mathbf{q}_2)}$ and $\sigma_1 \models \mathbf{q}_1 \wedge \mathbf{q}_2$, we know $\sigma_1 \models \mathbf{q}_1$ and $\sigma_2 \models \mathbf{q}_2$. From $fv(\mathbf{q}_1 \wedge \mathbf{q}_2) = fv(\mathbf{q}_1) \cup fv(\mathbf{q}_2)$ we know $fv(\mathbf{q}_1) \subseteq fv(\mathbf{q}_1 \wedge \mathbf{q}_2)$. From $\sigma_1|_{fv(\mathbf{q}_1\wedge\mathbf{q}_2)} = \sigma_2|_{fv(\mathbf{q}_1\wedge\mathbf{q}_2)}$ by Lem. 262 we know $\sigma_1|_{fv(\mathbf{q}_1)} = \sigma_2|_{fv(\mathbf{q}_1)}$. From $\sigma_1 \models \mathbf{q}_1$ by IH1 we have $\sigma_2 \models \mathbf{q}_1$. Similarly we can prove $\sigma_2 \models \mathbf{q}_2$, thus $\sigma_2 \models \mathbf{q}_1 \wedge \mathbf{q}_2$.
- case $\mathbf{q}_1 \vee \mathbf{q}_2$.
  IH1: for all σ_1, σ_2, if $\sigma_1|_{fv(\mathbf{q}_1)} = \sigma_2|_{fv(\mathbf{q}_1)}$, then $\sigma_1 \models \mathbf{q}_1$ iff $\sigma_2 \models \mathbf{q}_1$.
  IH2: for all σ_1, σ_2, if $\sigma_1|_{fv(\mathbf{q}_2)} = \sigma_2|_{fv(\mathbf{q}_2)}$, then $\sigma_1 \models \mathbf{q}_2$ iff $\sigma_2 \models \mathbf{q}_2$.
  For all σ_1, σ_2 such that $\sigma_1|_{fv(\mathbf{q}_1\vee\mathbf{q}_2)} = \sigma_2|_{fv(\mathbf{q}_1\vee\mathbf{q}_2)}$ and $\sigma_1 \models \mathbf{q}_1 \vee \mathbf{q}_2$, we know $\sigma_1 \models \mathbf{q}_1$ or $\sigma_1 \models \mathbf{q}_2$. We only prove the case $\sigma_1 \models \mathbf{q}_1$. The other case is similar. From $fv(\mathbf{q}_1 \vee \mathbf{q}_2) = fv(\mathbf{q}_1) \cup fv(\mathbf{q}_2)$ we know $fv(\mathbf{q}_1) \subseteq fv(\mathbf{q}_1 \vee \mathbf{q}_2)$. From $\sigma_1|_{fv(\mathbf{q}_1\vee\mathbf{q}_2)} = \sigma_2|_{fv(\mathbf{q}_1\vee\mathbf{q}_2)}$ by Lem. 262 we know $\sigma_1|_{fv(\mathbf{q}_1)} = \sigma_2|_{fv(\mathbf{q}_1)}$. From $\sigma_1 \models \mathbf{q}_1$ by IH1 we have $\sigma_2 \models \mathbf{q}_1$, thus $\sigma_2 \models \mathbf{q}_1 \vee \mathbf{q}_2$.
- case $\forall X.\mathbf{q}$.
  IH: for all σ_1, μ_2, if $\sigma_1|_{fv(\mathbf{q})} = \sigma_2|_{fv(\mathbf{q})}$, then $\sigma_1 \models \mathbf{q}$ iff $\sigma_2 \models \mathbf{q}$.
  For all σ_1, σ_2 such that $\sigma_1|_{fv(\forall X.\mathbf{q})} = \sigma_2|_{fv(\forall X.\mathbf{q})}$ and $\sigma_1 \models \forall X.\mathbf{q}$, from $fv(\forall X.\mathbf{q}) = fv(\mathbf{q}) - \{X\}$ we know $\sigma_1|_{fv(\mathbf{q})-\{X\}} = \sigma_2|_{fv(\mathbf{q})-\{X\}}$. To prove $\sigma_2 \models \forall X.\mathbf{q}$, we need to prove $\sigma_2\{X \rightsquigarrow r\} \models \mathbf{q}$ for all r. For all r, from $\sigma_1 \models \forall X.\mathbf{q}$ we know $\sigma_1\{X \rightsquigarrow r\} \models \mathbf{q}$. From $\sigma_1|_{fv(\mathbf{q})-\{X\}} = \sigma_2|_{fv(\mathbf{q})-\{X\}}$ by Lem. 254 we have $\sigma_1\{X \rightsquigarrow r\}|_{fv(\mathbf{q})} = \sigma_2\{X \rightsquigarrow r\}|_{fv(\mathbf{q})}$. From $\sigma_1\{X \rightsquigarrow r\} \models \mathbf{q}$ by IH we have $\sigma_2\{X \rightsquigarrow r\} \models \mathbf{q}$.
- case $\exists X.\mathbf{q}$.
  IH: for all σ_1, σ_2, if $\sigma_1|_{fv(\mathbf{q})} = \sigma_2|_{fv(\mathbf{q})}$, then $\sigma_1 \models \mathbf{q}$ iff $\sigma_2 \models \mathbf{q}$.
  For all σ_1, σ_2 such that $\sigma_1|_{fv(\exists X.\mathbf{q})} = \sigma_2|_{fv(\exists X.\mathbf{q})}$ and $\sigma_1 \models \exists X.\mathbf{q}$, we know there exists r such that $\sigma_1\{X \rightsquigarrow r\} \models \mathbf{q}$. From $fv(\forall X.\mathbf{q}) = fv(\mathbf{q}) - \{X\}$ we know $\sigma_1|_{fv(\mathbf{q})-\{X\}} = \sigma_2|_{fv(\mathbf{q})-\{X\}}$. By Lem. 254 we have $\sigma_1\{X \rightsquigarrow r\}|_{fv(\mathbf{q})} = \sigma_2\{X \rightsquigarrow r\}|_{fv(\mathbf{q})}$. From $\sigma_1\{X \rightsquigarrow r\} \models \mathbf{q}$ by IH we have $\sigma_2\{X \rightsquigarrow r\} \models \mathbf{q}$, thus $\sigma_2 \models \exists X.\mathbf{q}$.

Definition 83. Let $\hat{\sigma} \in VS \to \mathbb{R}$, we define $pad(\hat{\sigma}) \stackrel{\text{def}}{=} \lambda x. \begin{cases} \hat{\sigma}(x), & \text{if } x \in VS \\ 0, & \text{otherwise} \end{cases}$.

Lemma 256. *For all VS and $\hat{\sigma} \in VS \to \mathbb{R}$, $(pad(\hat{\sigma}))|_{VS} = \hat{\sigma}$.*

Proof. For all VS and $\hat{\sigma} \in VS \to \mathbb{R}$, to prove $(pad(\hat{\sigma}))|_{VS} = \hat{\sigma}$, we need to prove $(pad(\hat{\sigma}))|_{VS}(x) = \hat{\sigma}(x)$ for all $x \in VS$. For all $x \in VS$, we have $(pad(\hat{\sigma}))|_{VS}(x) = pad(\hat{\sigma})(x) = \hat{\sigma}(x)$.

Lemma 257. *For all e and σ, $[\![e]\!]_{pad(\sigma|_{fv(e)})} = [\![e]\!]_{\sigma}$.*

Proof. For all e and σ, we know $\sigma|_{fv(e)} \in fv(e) \to \mathbb{R}$. By Lem. 256 we know $(pad(\sigma|_{fv(e)}))|_{fv(e)}$
$= \sigma|_{fv(e)}$. By Lem. 252 we have $[\![e]\!]_{pad(\sigma|_{fv(e)})} = [\![e]\!]_{\sigma}$.

Lemma 258. *For all $\mathbf{q}$ and σ, $pad(\sigma|_{fv(\mathbf{q})}) \models \mathbf{q}$ if and only if $\sigma \models \mathbf{q}$.*

Proof. For all $\mathbf{q}$ and σ, we know $\sigma|_{fv(\mathbf{q})} \in fv(\mathbf{q}) \to \mathbb{R}$. By Lem. 256 we know $(pad(\sigma|_{fv(\mathbf{q})}))|_{fv(\mathbf{q})}$
$= \sigma|_{fv(\mathbf{q})}$. By Lem. 255 we have $pad(\sigma|_{fv(\mathbf{q})}) \models \mathbf{q}$ if and only if $\sigma \models \mathbf{q}$.

Lemma 259. *For all ξ, μ_1, μ_2, if $\mu_1|_{fv(\xi)} = \mu_2|_{fv(\xi)}$, then $[\![\xi]\!]_{\mu_1} = [\![\xi]\!]_{\mu_2}$.*

Proof. by induction on ξ.

- case r.
  For all μ_1, μ_2 such that $\mu_1|_{fv(r)} = \mu_2|_{fv(r)}$, we have $[\![r]\!]_{\mu_1} = r = [\![r]\!]_{\mu_2}$.
- case $\mathbb{E}(e)$.
  For all μ_1, μ_2 such that $\mu_1|_{fv(\mathbb{E}(e))} = \mu_2|_{fv(\mathbb{E}(e))}$, from $fv(\mathbb{E}(e)) = fv(e)$ we have $\mu_1|_{fv(e)} = \mu_2|_{fv(e)}$. We have

$$
\begin{aligned}
& [\![\mathbb{E}(e)]\!]_{\mu_1} \\
&= \mathbb{E}_{\sigma\sim\mu_1}[[\![e]\!]_{\sigma}] \\
&= \textstyle\sum_{\sigma} \mu_1(\sigma) \cdot [\![e]\!]_{\sigma} \\
&= \textstyle\sum_{\sigma} \mu_1(\sigma) \cdot [\![e]\!]_{pad(\sigma|_{fv(e)})} \qquad \text{(by Lem. 257)} \\
&= \textstyle\sum_{\sigma} \sum_{\hat{\sigma}\in fv(e)\to\mathbb{R}} \{\mu_1(\sigma) \cdot [\![e]\!]_{pad(\hat{\sigma})} \mid \sigma|_{fv(e)} = \hat{\sigma}\} \\
&= \textstyle\sum_{\hat{\sigma}\in fv(e)\to\mathbb{R}} [\![e]\!]_{pad(\hat{\sigma})} \cdot \sum_{\sigma} \{\mu_1(\sigma) \mid \sigma|_{fv(e)} = \hat{\sigma}\} \\
&= \textstyle\sum_{\hat{\sigma}\in fv(e)\to\mathbb{R}} [\![e]\!]_{pad(\hat{\sigma})} \cdot \mu_1|_{fv(e)}.
\end{aligned}
$$

  Similarly we can prove $[\![\mathbb{E}(e)]\!]_{\mu_2} = \sum_{\hat{\sigma}\in fv(e)\to\mathbb{R}} [\![e]\!]_{pad(\hat{\sigma})} \cdot \mu_2|_{fv(e)}$. From $\mu_1|_{fv(e)} = \mu_2|_{fv(e)}$ we have $[\![\mathbb{E}(e)]\!]_{\mu_1} = [\![\mathbb{E}(e)]\!]_{\mu_2}$.
- case $\mathbf{Pr}(\mathbf{q})$.
  For all μ_1, μ_2 such that $\mu_1|_{fv(\mathbf{Pr}(\mathbf{q}))} = \mu_2|_{fv(\mathbf{Pr}(\mathbf{q}))}$, from $fv(\mathbf{Pr}(\mathbf{q})) = fv(\mathbf{q})$ we have $\mu_1|_{fv(\mathbf{q})} = \mu_2|_{fv(\mathbf{q})}$. We have

$$
\begin{aligned}
& [\![\mathbf{Pr}(\mathbf{q})]\!]_{\mu_1} \\
&= \mathbf{Pr}_{\sigma\sim\mu_1}[\sigma \models \mathbf{q}] \\
&= \textstyle\sum_{\sigma} \{\mu_1(\sigma) \mid \sigma \models \mathbf{q}\} \\
&= \textstyle\sum_{\sigma} \{\mu_1(\sigma) \mid pad(\sigma|_{fv(\mathbf{q})}) \models \mathbf{q}\} \qquad \text{(by Lem. 258)} \\
&= \textstyle\sum_{\sigma} \sum_{\hat{\sigma}\in fv(e)\to\mathbb{R}} \{\mu_1(\sigma) \mid \sigma|_{fv(\mathbf{q})} = \hat{\sigma} \wedge pad(\hat{\sigma}) \models \mathbf{q}\} \\
&= \textstyle\sum_{\hat{\sigma}\in fv(e)\to\mathbb{R}} \{pad(\hat{\sigma}) \models \mathbf{q}\} \cdot \sum_{\sigma} \{\sigma|_{fv(\mathbf{q})} = \hat{\sigma}\} \\
&= \textstyle\sum_{\hat{\sigma}\in fv(e)\to\mathbb{R}} \{pad(\hat{\sigma}) \models \mathbf{q}\} \cdot \mu_1|_{fv(\mathbf{q})}.
\end{aligned}
$$

  Similarly we can prove $[\![\mathbf{Pr}(\mathbf{q})]\!]_{\mu_2} = \sum_{\hat{\sigma}\in fv(e)\to\mathbb{R}} \{pad(\hat{\sigma}) \models \mathbf{q}\} \cdot \mu_2|_{fv(\mathbf{q})}$. From $\mu_1|_{fv(\mathbf{q})} = \mu_2|_{fv(\mathbf{q})}$ we have $[\![\mathbf{Pr}(\mathbf{q})]\!]_{\mu_1} = [\![\mathbf{Pr}(\mathbf{q})]\!]_{\mu_2}$.

- case $\xi_1 + \xi_2$.
  IH1: For all μ_1, μ_2, if $\mu_1|_{fv(\xi_1)} = \mu_2|_{fv(\xi_1)}$, then $[\![\xi_1]\!]_{\mu_1} = [\![\xi_1]\!]_{\mu_2}$.
  IH1: For all μ_1, μ_2, if $\mu_1|_{fv(\xi_2)} = \mu_2|_{fv(\xi_2)}$, then $[\![\xi_2]\!]_{\mu_1} = [\![\xi_2]\!]_{\mu_2}$.
  For all μ_1, μ_2 such that $\mu_1|_{fv(\xi_1+\xi_2)} = \mu_2|_{fv(\xi_1+\xi_2)}$, from $fv(\xi_1 + \xi_2) = fv(\xi_1) \cup fv(\xi_2)$ we know $fv(\xi_1) \subseteq fv(\xi_1+\xi_2)$. From $\mu_1|_{fv(\xi_1+\xi_2)} = \mu_2|_{fv(\xi_1+\xi_2)}$ by Lem. 262 we know $\mu_1|_{fv(\xi_1)} = \mu_2|_{fv(\xi_1)}$. By IH1 we have $[\![\xi_1]\!]_{\mu_1} = [\![\xi_1]\!]_{\mu_2}$. Similarly we can prove $[\![\xi_2]\!]_{\mu_1} = [\![\xi_2]\!]_{\mu_2}$. Therefore $[\![\xi_1 + \xi_2]\!]_{\mu_1} = [\![\xi_1]\!]_{\mu_1} + [\![\xi_2]\!]_{\mu_1} = [\![\xi_1]\!]_{\mu_2} + [\![\xi_2]\!]_{\mu_2} = [\![\xi_1 + \xi_2]\!]_{\mu_2}$.
- case $\xi_1 - \xi_2$.
  Similar to the case $\xi_1 + \xi_2$.
- case $\xi_1 * \xi_2$.
  Similar to the case $\xi_1 + \xi_2$.

Lemma 260. *For all VS, VS', σ, if $VS' \subseteq VS$, then $(\sigma|_{VS})|_{VS'} = \sigma|_{VS'}$.*

Proof. For all VS, VS', σ such that $VS' \subseteq VS$, to prove $(\sigma|_{VS})|_{VS'} = \sigma|_{VS'}$, we need to prove $(\sigma|_{VS})|_{VS'}(x) = \sigma|_{VS'}(x)$ for all $x \in VS'$. For all $x \in VS'$, from $VS' \subseteq VS$ we know $x \in VS$, thus $(\sigma|_{VS})|_{VS'}(x) = \sigma|_{VS}(x) = \sigma(x) = \sigma|_{VS'}(x)$.

Lemma 261. *For all VS, VS', μ, if $VS' \subseteq VS$, then $(\mu|_{VS})|_{VS'} = \mu|_{VS'}$.*

Proof. For all VS, VS', μ such that $VS' \subseteq VS$, by Lem. 260 we know $(\sigma|_{VS})|_{VS'} = \sigma|_{VS'}$ for all σ, thus

$$\begin{aligned}
&(\mu|_{VS})|_{VS'} \\
&= \lambda\hat{\sigma} \in VS' \to \mathbb{R}. \textstyle\sum_{\bar{\sigma} \in VS \to \mathbb{R}} \{\mu|_{VS}(\bar{\sigma}) \mid \bar{\sigma}|_{VS'} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS' \to \mathbb{R}. \textstyle\sum_{\bar{\sigma} \in VS \to \mathbb{R}} \{\sum_{\sigma} \{\mu(\sigma) \mid \sigma|_{VS} = \bar{\sigma}\} \mid \bar{\sigma}|_{VS'} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS' \to \mathbb{R}. \textstyle\sum_{\sigma} \sum_{\bar{\sigma} \in VS \to \mathbb{R}} \{\mu(\sigma) \mid \sigma|_{VS} = \bar{\sigma} \wedge \bar{\sigma}|_{VS'} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS' \to \mathbb{R}. \textstyle\sum_{\sigma} \{\mu(\sigma) \mid (\sigma|_{VS})|_{VS'} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS' \to \mathbb{R}. \textstyle\sum_{\sigma} \{\mu(\sigma) \mid \sigma|_{VS'} = \hat{\sigma}\} \\
&= \mu|_{VS'}.
\end{aligned}$$

Lemma 262. *For all VS, VS', μ_1, μ_2, if $VS' \subseteq VS$ and $\mu_1|_{VS} = \mu_2|_{VS}$, then $\mu_1|_{VS'} = \mu_2|_{VS'}$.*

Proof. For all VS, VS', μ_1, μ_2 such that $VS' \subseteq VS$ and $\mu_1|_{VS} = \mu_2|_{VS}$, by Lem. 261 we know $\mu_1|_{VS'} = (\mu_1|_{VS})|_{VS'} = (\mu_2|_{VS})|_{VS'} = \mu_2|_{VS'}$.

Lemma 263. *For all VS, X, r, σ, if $X \in VS$, then $\sigma\{X \rightsquigarrow r\}|_{VS} = \sigma|_{VS-\{X\}}\{X \rightsquigarrow r\}$.*

Proof. For all VS, X, r, σ such that $X \in VS$, to prove $\sigma\{X \rightsquigarrow r\}|_{VS} = \sigma|_{VS-\{X\}}\{X \rightsquigarrow r\}$, we need to prove $\sigma\{X \rightsquigarrow r\}|_{VS}(x) = \sigma|_{VS-\{X\}}\{X \rightsquigarrow r\}(x)$ for all $x \in VS$. For all $x \in VS$, if $x = X$, then $\sigma\{X \rightsquigarrow r\}|_{VS}(x) = \sigma\{X \rightsquigarrow r\}(x) = r = \sigma|_{VS-\{X\}}\{X \rightsquigarrow r\}(x)$. Otherwise $x \neq X$, then $x \in VS - \{X\}$, thus $\sigma\{X \rightsquigarrow r\}|_{VS}(x) = \sigma\{X \rightsquigarrow r\}(x) = \sigma(x) = \sigma|_{VS-\{X\}}(x) = \sigma|_{VS-\{X\}}\{X \rightsquigarrow r\}(x)$.

Lemma 264. *For all VS, X, r, μ, if $X \in VS$, then $\mu\{X \rightsquigarrow r\}|_{VS} = \mu|_{VS-\{X\}}\{X \rightsquigarrow r\}$.*

Proof. From $X \in VS$ by Lem. 263 we know $\sigma\{X \rightsquigarrow r\}|_{VS} = \sigma|_{VS-\{X\}}\{X \rightsquigarrow r\}$ for all σ, thus

$$
\begin{aligned}
&\mu\{X \rightsquigarrow r\}|_{VS} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma}\{\mu\{X \rightsquigarrow r\}(\sigma) \mid \sigma|_{VS} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma}\{\sum_{\sigma'}\{\mu(\sigma') \mid \sigma'\{X \rightsquigarrow r\} = \sigma\} \mid \sigma|_{VS} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma,\sigma'}\{\mu(\sigma') \mid \sigma'\{X \rightsquigarrow r\} = \sigma \wedge \sigma|_{VS} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma'}\{\mu(\sigma') \mid \sigma'\{X \rightsquigarrow r\}|_{VS} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma'}\{\mu(\sigma') \mid \sigma'|_{VS-\{X\}}\{X \rightsquigarrow r\} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma'}\sum_{\bar{\sigma}\in(VS-\{X\})\to\mathbb{R}}\{\mu(\sigma') \mid \sigma'|_{VS-\{X\}} = \sigma \wedge \sigma\{X \rightsquigarrow r\} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\bar{\sigma}\in(VS-\{X\})\to\mathbb{R}}\{\sum_{\sigma'}\{\mu(\sigma') \mid \sigma'|_{VS-\{X\}} = \sigma\} \mid \sigma\{X \rightsquigarrow r\} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\bar{\sigma}\in(VS-\{X\})\to\mathbb{R}}\{\mu|_{VS-\{X\}}(\sigma) \mid \sigma\{X \rightsquigarrow r\} = \hat{\sigma}\} \\
&= \mu|_{VS-\{X\}}\{X \rightsquigarrow r\}.
\end{aligned}
$$

Lemma 265. *For all VS, X, r, σ, if $X \notin VS$, then $\sigma\{X \rightsquigarrow r\}|_{VS} = \sigma|_{VS}$.*

Proof. For all VS, X, r, σ such that $X \notin VS$, to prove $\sigma\{X \rightsquigarrow r\}|_{VS} = \sigma|_{VS}$, we need to prove $\sigma\{X \rightsquigarrow r\}|_{VS}(x) = \sigma|_{VS}(x)$ for all $x \in VS$. For all $x \in VS$, from $X \notin VS$ we know $x \neq X$, thus $\sigma\{X \rightsquigarrow r\}|_{VS}(x) = \sigma\{X \rightsquigarrow r\}(x) = \sigma(x) = \sigma|_{VS}(x)$.

Lemma 266. *For all VS, X, r, μ, if $X \notin VS$, then $\mu\{X \rightsquigarrow r\}|_{VS} = \mu|_{VS}$.*

Proof. From $X \notin VS$ by Lem. 265 we know $\sigma\{X \rightsquigarrow r\}|_{VS} = \sigma|_{VS}$ for all σ, thus

$$
\begin{aligned}
&\mu\{X \rightsquigarrow r\}|_{VS} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma}\{\mu\{X \rightsquigarrow r\}(\sigma) \mid \sigma|_{VS} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma}\{\sum_{\sigma'}\{\mu(\sigma') \mid \sigma'\{X \rightsquigarrow r\} = \sigma\} \mid \sigma|_{VS} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma,\sigma'}\{\mu(\sigma') \mid \sigma'\{X \rightsquigarrow r\} = \sigma \wedge \sigma|_{VS} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma'}\{\mu(\sigma') \mid \sigma'\{X \rightsquigarrow r\}|_{VS} = \hat{\sigma}\} \\
&= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma'}\{\mu(\sigma') \mid \sigma'|_{VS} = \hat{\sigma}\} \\
&= \mu|_{VS}.
\end{aligned}
$$

Lemma 267. *For all VS, X, r, μ_1, μ_2, if $\mu_1|_{VS-\{X\}} = \mu_2|_{VS-\{X\}}$, then $\mu_1\{X \rightsquigarrow r\}|_{VS} = \mu_2\{X \rightsquigarrow r\}|_{VS}$.*

Proof. For all $VS, X, r, \sigma_1, \sigma_2$ such that $\mu_1|_{VS-\{X\}} = \mu_2|_{VS-\{X\}}$, there are two cases: $X \in VS$ or $X \notin VS$. We prove the two cases respectively.

- $X \in VS$.
  From $X \in VS$ by Lem. 264 we have $\mu_1\{X \rightsquigarrow r\}|_{VS} = \mu_1|_{VS-\{X\}}\{X \rightsquigarrow r\} = \mu_2|_{VS-\{X\}}\{X \rightsquigarrow r\} = \mu_2\{X \rightsquigarrow r\}|_{VS}$.
- $X \notin VS$.
  We have $VS - \{X\} = VS$. From $\mu_1|_{VS-\{X\}} = \mu_2|_{VS-\{X\}}$ we know $\mu_1|_{VS} = \mu_2|_{VS}$. From $X \notin VS$ by Lem. 266 we have $\mu_1\{X \rightsquigarrow r\}|_{VS} = \mu_1|_{VS} = \mu_2|_{VS} = \mu_2\{X \rightsquigarrow r\}|_{VS}$.

Lemma 268. *For all $p, \mu_1, \mu_2, \mu', VS$, if $(\mu_1 \oplus_p \mu_2)|_{VS} = \mu'|_{VS}$, then there exists μ'_1 and μ'_2 such that $\mu' = \mu'_1 \oplus_p \mu'_2$, $\mu_1|_{VS} = \mu'_1|_{VS}$ and $\mu_2|_{VS} = \mu'_2|_{VS}$.*

Proof. For all $p, \mu_1, \mu_2, \mu', VS$ such that $(\mu_1 \oplus_p \mu_2)|_{VS} = \mu'|_{VS}$, by Lem. 268 we know $(\mu_1 \oplus_p \mu_2)|_{VS}$ $= \mu_1|_{VS} \oplus_p \mu_2|_{VS}$, thus $\mu'|_{VS} = \mu_1|_{VS} \oplus_p \mu_2|_{VS}$. Let $\mu'_1 = \lambda\sigma. \begin{cases} \frac{\mu_1|_{VS}(\sigma|_{VS}) \cdot \mu'(\sigma)}{\mu'|_{VS}(\sigma|_{VS})}, & \text{if } \mu'|_{VS}(\sigma|_{VS}) > 0 \\ 0, & \text{otherwise} \end{cases}$ and $\mu'_2 = \lambda\sigma. \begin{cases} \frac{\mu_2|_{VS}(\sigma|_{VS}) \cdot \mu'(\sigma)}{\mu'|_{VS}(\sigma|_{VS})}, & \text{if } \mu'|_{VS}(\sigma|_{VS}) > 0 \\ 0, & \text{otherwise} \end{cases}$. To prove $\mu' = \mu'_1 \oplus_p \mu'_2$, we need to prove $\mu'(\sigma) = p \cdot \mu'_1(\sigma) + (1-p) \cdot \mu'_2(\sigma)$ for all σ. For all σ, if $\mu'|_{VS}(\sigma|_{VS}) > 0$, then

$$\begin{aligned} & p \cdot \mu'_1(\sigma) + (1-p) \cdot \mu'_2(\sigma) \\ = {} & p \cdot \tfrac{\mu_1|_{VS}(\sigma|_{VS}) \cdot \mu'(\sigma)}{\mu'|_{VS}(\sigma|_{VS})} + (1-p) \cdot \tfrac{\mu_2|_{VS}(\sigma|_{VS}) \cdot \mu'(\sigma)}{\mu'|_{VS}(\sigma|_{VS})} \\ = {} & \tfrac{p \cdot \mu_1|_{VS}(\sigma|_{VS}) + (1-p) \cdot \mu_2|_{VS}(\sigma|_{VS})}{\mu'|_{VS}(\sigma|_{VS})} \cdot \mu'(\sigma) \\ = {} & \tfrac{(\mu_1|_{VS} \oplus_p \mu_2|_{VS})(\sigma|_{VS})}{\mu'|_{VS}(\sigma|_{VS})} \cdot \mu'(\sigma) \\ = {} & \tfrac{\mu'|_{VS}(\sigma|_{VS})}{\mu'|_{VS}(\sigma|_{VS})} \cdot \mu'(\sigma) \\ = {} & \mu'(\sigma). \end{aligned}$$

Otherwise $\mu'|_{VS}(\sigma|_{VS}) = 0$, then $\mu'_1(\sigma) = \mu'_2(\sigma) = 0$. From $0 = \mu'|_{VS}(\sigma|_{VS}) = \sum_{\sigma'}\{\mu'(\sigma') \mid \sigma'|_{VS} = \sigma|_{VS}\} \geq \mu'(\sigma)$ we know $\mu'(\sigma) = 0$, thus $\mu'(\sigma) = p \cdot \mu'_1(\sigma) + (1-p) \cdot \mu'_2(\sigma)$. From $\forall\sigma.\ \mu'|_{VS}(\sigma|_{VS}) \geq \mu'(\sigma)$ we know $\forall\sigma.\ \mu'(\sigma) > 0 \implies \mu'|_{VS}(\sigma|_{VS}) > 0$, thus

$$\begin{aligned} & \mu'_1|_{VS} \\ = {} & \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_\sigma\{\mu'_1(\sigma) \mid \sigma|_{VS} = \hat{\sigma}\} \\ = {} & \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_\sigma\{\frac{\mu_1|_{VS}(\sigma|_{VS}) \cdot \mu'(\sigma)}{\mu'|_{VS}(\sigma|_{VS})} \mid \sigma|_{VS} = \hat{\sigma} \wedge \mu'|_{VS}(\sigma|_{VS}) > 0\} \\ = {} & \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_\sigma\{\frac{\mu_1|_{VS}(\hat{\sigma}) \cdot \mu'(\sigma)}{\mu'|_{VS}(\hat{\sigma})} \mid \sigma|_{VS} = \hat{\sigma} \wedge \mu'|_{VS}(\sigma|_{VS}) > 0\} \\ = {} & \lambda\hat{\sigma} \in VS \to \mathbb{R}. \tfrac{\mu_1|_{VS}(\hat{\sigma})}{\mu'|_{VS}(\hat{\sigma})} \cdot \textstyle\sum_\sigma\{\mu'(\sigma) \mid \sigma|_{VS} = \hat{\sigma} \wedge \mu'|_{VS}(\sigma|_{VS}) > 0\} \\ = {} & \lambda\hat{\sigma} \in VS \to \mathbb{R}. \tfrac{\mu_1|_{VS}(\hat{\sigma})}{\mu'|_{VS}(\hat{\sigma})} \cdot \textstyle\sum_\sigma\{\mu'(\sigma) \mid \sigma|_{VS} = \hat{\sigma} \wedge \mu'|_{VS}(\sigma|_{VS}) > 0 \wedge \mu'(\sigma) > 0\} \\ = {} & \lambda\hat{\sigma} \in VS \to \mathbb{R}. \tfrac{\mu_1|_{VS}(\hat{\sigma})}{\mu'|_{VS}(\hat{\sigma})} \cdot \textstyle\sum_\sigma\{\mu'(\sigma) \mid \sigma|_{VS} = \hat{\sigma} \wedge \mu'(\sigma) > 0\} \\ = {} & \lambda\hat{\sigma} \in VS \to \mathbb{R}. \tfrac{\mu_1|_{VS}(\hat{\sigma})}{\mu'|_{VS}(\hat{\sigma})} \cdot \textstyle\sum_\sigma\{\mu'(\sigma) \mid \sigma|_{VS} = \hat{\sigma}\} \\ = {} & \lambda\hat{\sigma} \in VS \to \mathbb{R}. \tfrac{\mu_1|_{VS}(\hat{\sigma})}{\mu'|_{VS}(\hat{\sigma})} \cdot \mu'|_{VS}(\hat{\sigma}) \\ = {} & \lambda\hat{\sigma} \in VS \to \mathbb{R}. \mu_1|_{VS}(\hat{\sigma}) \\ = {} & \mu_1|_{VS}. \end{aligned}$$

Similarly we can prove $\mu'_2|_{VS} = \mu_2|_{VS}$.

Definition 84. Let $V \in \mathbb{D}_{\mathbb{D}_{State}}$, we define $V|_{VS} \stackrel{\text{def}}{=} \lambda\hat{\mu} \in \mathbb{D}_{VS \to \mathbb{R}}. \sum_\mu\{V(\mu) \mid \mu|_{VS} = \hat{\mu}\}$.

Lemma 269. *For all $V \in \mathbb{D}_{\mathbb{D}_{State}}$ and VS, $\overline{V}|_{VS} = \lambda\hat{\sigma} \in VS \to \mathbb{R}. \sum_\nu V(\nu) \cdot \nu|_{VS}(\hat{\sigma})$.*

Proof. For all $V \in \mathbb{D}_{\mathbb{D}_{State}}$ and VS, we have

$$\begin{aligned}\overline{V}|_{VS} &= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma}\{\overline{V}(\sigma) \mid \sigma|_{VS} = \hat{\sigma}\} \\ &= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma,\nu}\{V(\nu)\cdot\nu(\sigma) \mid \sigma|_{VS} = \hat{\sigma}\} \\ &= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\nu} V(\nu)\cdot\sum_{\sigma}\{\nu(\sigma) \mid \sigma|_{VS} = \hat{\sigma}\} \\ &= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\nu} V(\nu)\cdot\nu|_{VS}(\hat{\sigma})\end{aligned}$$

Definition 85. $zoom(\mu, \mu', VS) \stackrel{\text{def}}{=} \lambda\sigma.\ \frac{\mu(\sigma)\cdot\mu'|_{VS}(\sigma|_{VS})}{\mu|_{VS}(\sigma|_{VS})}$.

Lemma 270. *For all* μ, μ', VS, $zoom(\mu, \mu', VS)|_{VS} = \mu'|_{VS}$.

Proof. For all μ, μ', VS, we have

$$\begin{aligned}&zoom(\mu, \mu', VS)|_{VS} \\ &= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma}\{zoom(\mu, \mu')(\sigma) \mid \sigma|_{VS} = \hat{\sigma}\} \\ &= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma}\{\frac{\mu(\sigma)\cdot\mu'|_{VS}(\sigma|_{VS})}{\mu|_{VS}(\sigma|_{VS})} \mid \sigma|_{VS} = \hat{\sigma}\} \\ &= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \textstyle\sum_{\sigma}\{\frac{\mu(\sigma)\cdot\mu'|_{VS}(\hat{\sigma})}{\mu|_{VS}(\hat{\sigma})} \mid \sigma|_{VS} = \hat{\sigma}\} \\ &= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \tfrac{\mu'|_{VS}(\hat{\sigma})}{\mu|_{VS}(\hat{\sigma})}\cdot\textstyle\sum_{\sigma}\{\mu(\sigma) \mid \sigma|_{VS} = \hat{\sigma}\} \\ &= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \tfrac{\mu'|_{VS}(\hat{\sigma})}{\mu|_{VS}(\hat{\sigma})}\cdot\mu|_{VS}(\hat{\sigma}) \\ &= \lambda\hat{\sigma} \in VS \to \mathbb{R}. \mu'|_{VS}(\hat{\sigma}) \\ &= \mu'|_{VS}.\end{aligned}$$

Lemma 271. *For all* $V \in \mathbb{D}_{\mathbb{D}_{State}}$ *and* VS, *if* $\overline{V}|_{VS} = \mu|_{VS}$, *then there exists* $V' \in \mathbb{D}_{\mathbb{D}_{State}}$ *such that* $\mu = \overline{V'}$ *and* $V|_{VS} = V'|_{VS}$.

Proof. For all $V \in \mathbb{D}_{\mathbb{D}_{State}}$ and VS such that $\overline{V}|_{VS} = \mu|_{VS}$, let $V' \stackrel{\text{def}}{=} \lambda\nu'. \sum_{\nu}\{V(\nu) \mid zoom(\mu, \nu) = \nu'\}$, then

$$\begin{aligned}\overline{V'} &= \lambda\sigma. \textstyle\sum_{\nu'} V'(\nu')\cdot\nu'(\sigma) \\ &= \lambda\sigma. \textstyle\sum_{\nu'}\sum_{\nu}\{V(\nu)\cdot\nu'(\sigma) \mid zoom(\mu, \nu) = \nu'\} \\ &= \lambda\sigma. \textstyle\sum_{\nu} V(\nu)\cdot zoom(\mu, \nu)(\sigma) \\ &= \lambda\sigma. \textstyle\sum_{\nu} V(\nu)\cdot\frac{\mu(\sigma)\cdot\nu|_{VS}(\sigma|_{VS})}{\mu|_{VS}(\sigma|_{VS})} \\ &= \lambda\sigma. \tfrac{\mu(\sigma)}{\mu|_{VS}(\sigma|_{VS})}\cdot\textstyle\sum_{\nu} V(\nu)\cdot\nu|_{VS}(\sigma|_{VS}) \\ &= \lambda\sigma. \tfrac{\mu(\sigma)}{\mu|_{VS}(\sigma|_{VS})}\cdot\overline{V}|_{VS}(\sigma|_{VS}) \qquad \text{(by Lem. 269)} \\ &= \lambda\sigma. \tfrac{\mu(\sigma)}{\mu|_{VS}(\sigma|_{VS})}\cdot\mu|_{VS}(\sigma|_{VS}) \\ &= \mu\end{aligned}$$

and

$$\begin{aligned}V'|_{VS} &= \lambda\hat{\nu} \in \mathbb{D}_{VS\to\mathbb{R}}. \textstyle\sum_{\nu'}\{V'(\nu') \mid \nu'|_{VS} = \hat{\nu}\} \\ &= \lambda\hat{\nu} \in \mathbb{D}_{VS\to\mathbb{R}}. \textstyle\sum_{\nu'}\sum_{\nu}\{V(\nu) \mid zoom(\mu, \nu) = \nu' \wedge \nu'|_{VS} = \hat{\nu}\} \\ &= \lambda\hat{\nu} \in \mathbb{D}_{VS\to\mathbb{R}}. \textstyle\sum_{\nu}\{V(\nu) \mid zoom(\mu, \nu)|_{VS} = \hat{\nu}\} \\ &= \lambda\hat{\nu} \in \mathbb{D}_{VS\to\mathbb{R}}. \textstyle\sum_{\nu}\{V(\nu) \mid \nu|_{VS} = \hat{\nu}\} \qquad \text{(by Lem. 270)} \\ &= V|_{VS}.\end{aligned}$$

Lemma 272. *For all Q, μ_1, μ_2, if $\mu_1|_{fv(Q)} = \mu_2|_{fv(Q)}$, then $\mu_1 \models Q$ if and only if $\mu_2 \models Q$.*

Proof. by induction on Q. We only prove one direction (if $\mu_1 \models Q$ then $\mu_2 \models Q$) in each case. The other direction is similar.

- case $\lceil \mathbf{q} \rceil$.
  For all μ_1, μ_2 such that $\mu_1|_{fv(\lceil \mathbf{q} \rceil)} = \mu_2|_{fv(\lceil \mathbf{q} \rceil)}$ and $\mu_1 \models \lceil \mathbf{q} \rceil$, from $fv(\lceil \mathbf{q} \rceil) = fv(\mathbf{q})$ we have $\mu_1|_{fv(\mathbf{q})} = \mu_2|_{fv(\mathbf{q})}$. To prove $\mu_2 \models \lceil \mathbf{q} \rceil$, we need to prove $\sigma \models \mathbf{q}$ for all $\sigma \in \mathit{supp}(\mu_2)$. For all $\sigma \in \mathit{supp}(\mu_2)$, we have $\mu_2(\sigma) > 0$, thus $\mu_2|_{fv(\mathbf{q})}(\sigma|_{fv(\mathbf{q})}) = \sum_{\sigma'}\{\mu_2(\sigma') \mid \sigma'|_{fv(\mathbf{q})} = \sigma|_{fv(\mathbf{q})}\} \geq \mu_2(\sigma) > 0$. From $\mu_1|_{fv(\mathbf{q})} = \mu_2|_{fv(\mathbf{q})}$ we know $0 < \mu_1|_{fv(\mathbf{q})}(\sigma|_{fv(\mathbf{q})}) = \sum_{\sigma'}\{\mu_1(\sigma') \mid \sigma'|_{fv(\mathbf{q})} = \sigma|_{fv(\mathbf{q})}\}$, so there exists σ' such that $\mu_1(\sigma') > 0$ and $\sigma'|_{fv(\mathbf{q})} = \sigma|_{fv(\mathbf{q})}$, thus $\sigma' \in \mathit{supp}(\mu_1)$. From $\mu_1 \models \lceil \mathbf{q} \rceil$ we know $\sigma' \models \mathbf{q}$. From $\sigma'|_{fv(\mathbf{q})} = \sigma|_{fv(\mathbf{q})}$ by Lem. 255 we have $\sigma \models \mathbf{q}$.
- case $\xi_1 < \xi_2$.
  For all μ_1, μ_2 such that $\mu_1|_{fv(\xi_1<\xi_2)} = \mu_2|_{fv(\xi_1<\xi_2)}$ and $\mu_1 \models \xi_1 < \xi_2$, from $fv(\xi_1 < \xi_2) = fv(\xi_1) \cup fv(\xi_2)$ we know $fv(\xi_1) \subseteq fv(\xi_1 < \xi_2)$. From $\mu_1|_{fv(\xi_1<\xi_2)} = \mu_2|_{fv(\xi_1<\xi_2)}$ by Lem. 262 we know $\mu_1|_{fv(\xi_1)} = \mu_2|_{fv(\xi_1)}$. By Lem. 259 we have $[\![\xi_1]\!]_{\mu_1} = [\![\xi_1]\!]_{\mu_2}$. Similarly we can prove $[\![\xi_2]\!]_{\mu_1} = [\![\xi_2]\!]_{\mu_2}$. From $\mu_1 \models \xi_1 < \xi_2$ we know $[\![\xi_1]\!]_{\mu_1} < [\![\xi_2]\!]_{\mu_1}$, thus $[\![\xi_1]\!]_{\mu_2} = [\![\xi_1]\!]_{\mu_1} < [\![\xi_2]\!]_{\mu_1} = [\![\xi_2]\!]_{\mu_2}$.
- case $\xi_1 = \xi_2$.
  Similar to the case $\xi_1 < \xi_2$.
- case $\xi_1 \leq \xi_2$.
  Similar to the case $\xi_1 < \xi_2$.
- case $\neg Q$.
  IH: for all μ_1, μ_2, if $\mu_1|_{fv(Q)} = \mu_2|_{fv(Q)}$, then $\mu_1 \models Q$ iff $\mu_2 \models Q$.
  For all μ_1, μ_2 such that $\mu_1|_{fv(\neg Q)} = \mu_2|_{fv(\neg Q)}$ and $\mu_1 \models \neg Q$, from $fv(\neg Q) = fv(Q)$ we know $\mu_1|_{fv(Q)} = \mu_2|_{fv(Q)}$. From $\mu_1 \models \neg Q$ we know $\mu_1 \models Q$ does not hold. By IH we know $\mu_2 \models Q$ does not hold, thus $\mu_2 \models \neg Q$.
- case $Q_1 \wedge Q_2$.
  IH1: for all μ_1, μ_2, if $\mu_1|_{fv(Q_1)} = \mu_2|_{fv(Q_1)}$, then $\mu_1 \models Q_1$ iff $\mu_2 \models Q_1$.
  IH2: for all μ_1, μ_2, if $\mu_1|_{fv(Q_2)} = \mu_2|_{fv(Q_2)}$, then $\mu_1 \models Q_2$ iff $\mu_2 \models Q_2$.
  For all μ_1, μ_2 such that $\mu_1|_{fv(Q_1 \wedge Q_2)} = \mu_2|_{fv(Q_1 \wedge Q_2)}$ and $\mu_1 \models Q_1 \wedge Q_2$, we know $\mu_1 \models Q_1$ and $\mu_2 \models Q_2$. From $fv(Q_1 \wedge Q_2) = fv(Q_1) \cup fv(Q_2)$ we know $fv(Q_1) \subseteq fv(Q_1 \wedge Q_2)$. From $\mu_1|_{fv(Q_1 \wedge Q_2)} = \mu_2|_{fv(Q_1 \wedge Q_2)}$ by Lem. 262 we know $\mu_1|_{fv(Q_1)} = \mu_2|_{fv(Q_1)}$. From $\mu_1 \models Q_1$ by IH1 we have $\mu_2 \models Q_1$. Similarly we can prove $\mu_2 \models Q_2$, thus $\mu_2 \models Q_1 \wedge Q_2$.
- case $Q_1 \vee Q_2$.
  IH1: for all μ_1, μ_2, if $\mu_1|_{fv(Q_1)} = \mu_2|_{fv(Q_1)}$, then $\mu_1 \models Q_1$ iff $\mu_2 \models Q_1$.
  IH2: for all μ_1, μ_2, if $\mu_1|_{fv(Q_2)} = \mu_2|_{fv(Q_2)}$, then $\mu_1 \models Q_2$ iff $\mu_2 \models Q_2$.
  For all μ_1, μ_2 such that $\mu_1|_{fv(Q_1 \vee Q_2)} = \mu_2|_{fv(Q_1 \vee Q_2)}$ and $\mu_1 \models Q_1 \vee Q_2$, we know $\mu_1 \models Q_1$ or $\mu_1 \models Q_2$. We only prove the case $\mu_1 \models Q_1$. The other case is similar. From $fv(Q_1 \vee Q_2) = fv(Q_1) \cup fv(Q_2)$ we know $fv(Q_1) \subseteq fv(Q_1 \vee Q_2)$. From $\mu_1|_{fv(Q_1 \vee Q_2)} = \mu_2|_{fv(Q_1 \vee Q_2)}$ by Lem. 262 we know $\mu_1|_{fv(Q_1)} = \mu_2|_{fv(Q_1)}$. From $\mu_1 \models Q_1$ by IH1 we have $\mu_2 \models Q_1$, thus $\mu_2 \models Q_1 \vee Q_2$.

- case $\forall X.Q$.
  IH: for all μ_1, μ_2, if $\mu_1|_{fv(Q)} = \mu_2|_{fv(Q)}$, then $\mu_1 \models Q$ iff $\mu_2 \models Q$.
  For all μ_1, μ_2 such that $\mu_1|_{fv(\forall X.Q)} = \mu_2|_{fv(\forall X.Q)}$ and $\mu_1 \models \forall X.Q$, from $fv(\forall X.Q) = fv(Q) - \{X\}$ we know $\mu_1|_{fv(Q)-\{X\}} = \mu_2|_{fv(Q)-\{X\}}$. To prove $\mu_2 \models \forall X.Q$, we need to prove $\mu_2\{X \rightsquigarrow r\} \models Q$ for all r. For all r, from $\mu_1 \models \forall X.Q$ we know $\mu_1\{X \rightsquigarrow r\} \models Q$. From $\mu_1|_{fv(Q)-\{X\}} = \mu_2|_{fv(Q)-\{X\}}$ by Lem. 267 we have $\mu_1\{X \rightsquigarrow r\}|_{fv(Q)} = \mu_2\{X \rightsquigarrow r\}|_{fv(Q)}$. From $\mu_1\{X \rightsquigarrow r\} \models Q$ by IH we have $\mu_2\{X \rightsquigarrow r\} \models Q$.
- case $\exists X.Q$.
  IH: for all μ_1, μ_2, if $\mu_1|_{fv(Q)} = \mu_2|_{fv(Q)}$, then $\mu_1 \models Q$ iff $\mu_2 \models Q$.
  For all μ_1, μ_2 such that $\mu_1|_{fv(\exists X.Q)} = \mu_2|_{fv(\exists X.Q)}$ and $\mu_1 \models \exists X.Q$, we know there exists r such that $\mu_1\{X \rightsquigarrow r\} \models Q$. From $fv(\forall X.Q) = fv(Q) - \{X\}$ we know $\mu_1|_{fv(Q)-\{X\}} = \mu_2|_{fv(Q)-\{X\}}$. By Lem. 267 we have $\mu_1\{X \rightsquigarrow r\}|_{fv(Q)} = \mu_2\{X \rightsquigarrow r\}|_{fv(Q)}$. From $\mu_1\{X \rightsquigarrow r\} \models Q$ by IH we have $\mu_2\{X \rightsquigarrow r\} \models Q$, thus $\mu_2 \models \exists X.Q$.
- case $Q_1 \oplus_p Q_2$.
  IH1: for all μ_1, μ_2, if $\mu_1|_{fv(Q_1)} = \mu_2|_{fv(Q_1)}$, then $\mu_1 \models Q_1$ iff $\mu_2 \models Q_1$.
  IH2: for all μ_1, μ_2, if $\mu_1|_{fv(Q_2)} = \mu_2|_{fv(Q_2)}$, then $\mu_1 \models Q_2$ iff $\mu_2 \models Q_2$.
  For all μ_1, μ_2 such that $\mu_1|_{fv(Q_1 \oplus_p Q_2)} = \mu_2|_{fv(Q_1 \oplus_p Q_2)}$ and $\mu_1 \models Q_1 \oplus_p Q_2$, from $fv(Q_1 \wedge Q_2) = fv(Q_1) \cup fv(Q_2)$ we know $\mu_1|_{fv(Q_1) \cup fv(Q_2)} = \mu_2|_{fv(Q_1) \cup fv(Q_2)}$. From $\mu_1 \models Q_1 \oplus_p Q_2$ we know there are three cases. We prove the three cases respectively.
  - $p = 1$ and $\mu_1 \models Q_1$.
    From $\mu_1|_{fv(Q_1) \cup fv(Q_2)} = \mu_2|_{fv(Q_1) \cup fv(Q_2)}$ and $fv(Q_1) \subseteq fv(Q_1) \cup fv(Q_2)$ by Lem. 262 we know $\mu_1|_{fv(Q_1)} = \mu_2|_{fv(Q_1)}$. From $\mu_1 \models Q_1$ by IH1 we have $\mu_2 \models Q_1$. From $p = 1$ we know $\mu_2 \models Q_1 \oplus_p Q_2$.
  - $p = 0$ and $\mu_1 \models Q_2$.
    From $\mu_1|_{fv(Q_1) \cup fv(Q_2)} = \mu_2|_{fv(Q_1) \cup fv(Q_2)}$ and $fv(Q_2) \subseteq fv(Q_1) \cup fv(Q_2)$ by Lem. 262 we know $\mu_1|_{fv(Q_2)} = \mu_2|_{fv(Q_2)}$. From $\mu_1 \models Q_2$ by IH1 we have $\mu_2 \models Q_2$. From $p = 0$ we know $\mu_2 \models Q_1 \oplus_p Q_2$.
  - $0 < p < 1$ and there exists μ_{11} and μ_{12} such that $\mu = \mu_{11} \oplus_p \mu_{12}$, $\mu_{11} \models Q_1$ and $\mu_{12} \models Q_2$.
    From $\mu_1|_{fv(Q_1) \cup fv(Q_2)} = \mu_2|_{fv(Q_1) \cup fv(Q_2)}$ we know $(\mu_{11} \oplus_p \mu_{12})|_{fv(Q_1) \cup fv(Q_2)} = \mu_2|_{fv(Q_1) \cup fv(Q_2)}$. By Lem. 268 there exists μ_{21} and μ_{22} such that $\mu_2 = \mu_{21} \oplus_p \mu_{22}$, $\mu_{11}|_{fv(Q_1) \cup fv(Q_2)} = \mu_{21}|_{fv(Q_1) \cup fv(Q_2)}$ and $\mu_{12}|_{fv(Q_1) \cup fv(Q_2)} = \mu_{22}|_{fv(Q_1) \cup fv(Q_2)}$. From $\mu_{11}|_{fv(Q_1) \cup fv(Q_2)} = \mu_{21}|_{fv(Q_1) \cup fv(Q_2)}$ and $fv(Q_1) \subseteq fv(Q_1) \cup fv(Q_2)$ by Lem. 262 we know $\mu_{11}|_{fv(Q_1)} = \mu_{21}|_{fv(Q_1)}$. From $\mu_{11} \models Q_1$ by IH1 we have $\mu_{21} \models Q_1$. Similarly we can prove $\mu_{22} \models Q_2$. From $0 < p < 1$ and $\mu_2 = \mu_{21} \oplus_p \mu_{22}$ we know $\mu_2 \models Q_1 \oplus_p Q_2$.
- case $Q_1 \oplus Q_2$.
  IH1: for all μ_1, μ_2, if $\mu_1|_{fv(Q_1)} = \mu_2|_{fv(Q_1)}$, then $\mu_1 \models Q_1$ iff $\mu_2 \models Q_1$.
  IH2: for all μ_1, μ_2, if $\mu_1|_{fv(Q_2)} = \mu_2|_{fv(Q_2)}$, then $\mu_1 \models Q_2$ iff $\mu_2 \models Q_2$.
  For all μ_1, μ_2 such that $\mu_1|_{fv(Q_1 \oplus Q_2)} = \mu_2|_{fv(Q_1 \oplus Q_2)}$ and $\mu_1 \models Q_1 \oplus Q_2$, we know there exists p such that $\mu_1 \models Q_1 \oplus_p Q_2$. From $fv(Q_1 \oplus Q_2) = fv(Q_1) \cup fv(Q_2)$ we know $\mu_1|_{fv(Q_1) \cup fv(Q_2)} = \mu_2|_{fv(Q_1) \cup fv(Q_2)}$. To prove

$\mu_2 \models Q_1 \oplus Q_2$, it suffices to prove $\mu_2 \models Q_1 \oplus_p Q_2$. The rest of the proof is the same as the case $Q_1 \oplus_p Q_2$.

– case $\bigoplus Q$.
IH: for all μ_1, μ_2, if $\mu_1|_{fv(Q)} = \mu_2|_{fv(Q)}$, then $\mu_1 \models Q$ iff $\mu_2 \models Q$.
For all μ_1, μ_2 such that $\mu_1|_{fv(\bigoplus Q)} = \mu_2|_{fv(\bigoplus Q)}$ and $\mu_1 \models \bigoplus Q$, from $fv(\bigoplus Q) = fv(Q)$ we know $\mu_1|_{fv(Q)} = \mu_2|_{fv(Q)}$. From $\mu_1 \models \bigoplus Q$ we know there exists $V_1 \in \mathbb{D}_{\mathbb{D}_{State}}$ such that $\mu_1 = \overline{V_1}$ and $\nu \models Q$ for all $\nu \in \mathit{supp}(V_1)$. From $\mu_1|_{fv(Q)} = \mu_2|_{fv(Q)}$ we have $\overline{V_1}|_{fv(Q)} = \mu_2|_{fv(Q)}$. By Lem. 271 there exists $V_2 \in \mathbb{D}_{\mathbb{D}_{State}}$ such that $\mu_2 = \overline{V_2}$ and $V_1|_{fv(Q)} = V_2|_{fv(Q)}$. To prove $\mu_2 \models \bigoplus Q$, it suffices to prove $\nu \models Q$ for all $\nu \in \mathit{supp}(V_2)$. For all $\nu \in \mathit{supp}(V_2)$, we have $V_2(\nu) > 0$, thus $V_2|_{fv(Q)}(\nu|_{fv(Q)}) = \sum_\mu \{V_2(\mu) \mid \mu|_{fv(Q)} = \nu|_{fv(Q)}\} \geq V_2(\nu) > 0$. From $V_1|_{fv(Q)} = V_2|_{fv(Q)}$ we know $0 < V_2|_{fv(Q)}(\nu|_{fv(Q)}) = V_1|_{fv(Q)}(\nu|_{fv(Q)}) = \sum_\mu \{V_1(\mu) \mid \mu|_{fv(Q)} = \nu|_{fv(Q)}\}$, so there exists μ such that $V_1(\mu) > 0$ and $\mu|_{fv(Q)} = \nu|_{fv(Q)}$, thus $\mu \in \mathit{supp}(V_1)$. From $\nu \models Q$ for all $\nu \in \mathit{supp}(V_1)$ we know $\mu \models Q$. From $\mu|_{fv(Q)} = \nu|_{fv(Q)}$ by IH we have $\nu \models Q$.

Lemma 273. *For all η and R, if $\mathbf{Id} \Rightarrow R$, then $\eta \overset{R}{\mapsto} \eta$.*

Proof. For all η and R such that $\mathbf{Id} \Rightarrow R$, let $\psi \overset{\text{def}}{=} \{((C,\sigma),(C,\sigma)) \mid (C,\sigma) \in \mathit{supp}(\eta)\}$, then $\mathit{dom}(\psi) = \{(C,\sigma) \mid (C,\sigma) \in \mathit{supp}(\eta)\} = \mathit{supp}(\eta)$ and $\mathit{range}(\psi) = \{(C,\sigma) \mid (C,\sigma) \in \mathit{supp}(\eta)\} = \mathit{supp}(\eta)$. For all $((C,\sigma),(C',\sigma')) \in \psi$, we have $C' = C$ and $\sigma' = \sigma$, thus $(\sigma,\sigma') \models \mathbf{Id}$. From $\mathbf{Id} \Rightarrow R$ we know $(\sigma,\sigma') \models R$.

Lemma 274. *For all η, θ, η', R, if $\eta \rightsquigarrow (\theta, \eta')$ and $\theta \subseteq [\![R]\!]$, then $\delta(C_0) \otimes \eta^{(State)} \overset{R}{\mapsto} \delta(C_0) \otimes \eta'^{(State)}$.*

Proof. For all η, θ, η', R such that $\eta \rightsquigarrow (\theta, \eta')$ and $\theta \subseteq [\![R]\!]$, we know $\eta' = \lambda(C',\sigma').\sum_{C,\sigma}\{\eta(C,\sigma)\cdot p \mid (C,\sigma) \xrightarrow{p} (C,\sigma')\}$ and $\theta = \{(\sigma,\sigma') \mid \exists C, C'.\ \eta(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C,\sigma') \wedge p > 0\}$. Let $\psi \overset{\text{def}}{=} \{((C_0,\sigma),(C_0,\sigma')) \mid \exists C, C'.\ \eta(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}$, then

$$
\begin{aligned}
\mathit{dom}(\psi) &= \{(C,\sigma) \mid \exists C', \sigma'.\ ((C,\sigma),(C',\sigma')) \in \psi\} \\
&= \{(C_0,\sigma) \mid \exists C, C', \sigma'.\ \eta(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C,\sigma') \wedge p > 0\} \\
&= \{(C_0,\sigma) \mid \exists C.\ (C,\sigma) \in \mathit{supp}(\eta)\} \\
&= \{(C_0,\sigma) \mid \sigma \in \mathit{range}(\mathit{supp}(\eta))\} \\
&= \{(C_0,\sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\} \qquad \text{(by Lem. 22)} \\
&= \{(C_0,\sigma) \mid \eta^{(State)}(\sigma) > 0\} \\
&= \{(C_0,\sigma) \mid \delta(C_0)(C_0) \cdot \eta^{(State)}(\sigma) > 0\} \\
&= \{(C_0,\sigma) \mid (\delta(C_0) \otimes \eta^{(State)})(C_0,\sigma) > 0\} \\
&= \mathit{supp}(\delta(C_0) \otimes \eta^{(State)})
\end{aligned}
$$

and

$$\begin{aligned}
range(\psi) &= \{(C',\sigma') \mid \exists C,\sigma.\ ((C,\sigma),(C',\sigma')) \in \psi\} \\
&= \{(C_0,\sigma') \mid \exists C,C',\sigma.\ \eta(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C,\sigma') \wedge p > 0\} \\
&= \{(C_0,\sigma') \mid \exists C'.\textstyle\sum_{C,\sigma}\{\eta(C,\sigma)\cdot p \mid (C,\sigma) \xrightarrow{p} (C,\sigma')\} > 0\} \\
&= \{(C_0,\sigma') \mid \exists C'.\eta'(C',\sigma') > 0\} \\
&= \{(C_0,\sigma') \mid \exists C'.(C',\sigma') \in supp(\eta')\} \\
&= \{(C_0,\sigma') \mid \sigma \in range(supp(\eta'))\} \\
&= \{(C_0,\sigma') \mid \sigma \in supp(\eta'^{(State)})\} \qquad \text{(by Lem. 22)} \\
&= \{(C_0,\sigma') \mid \eta'^{(State)}(\sigma') > 0\} \\
&= \{(C_0,\sigma') \mid \delta(C_0)(C_0)\cdot\eta'^{(State)}(\sigma') > 0\} \\
&= \{(C_0,\sigma') \mid (\delta(C_0)\otimes\eta'^{(State)})(C_0,\sigma') > 0\} \\
&= supp(\delta(C_0)\otimes\eta'^{(State)}).
\end{aligned}$$

For all $((C,\sigma),(C',\sigma')) \in \psi$, we have $C' = C = C_0$ and there exists C, C' such that $\eta(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0$, thus $(\sigma,\sigma') \subseteq \theta$. From $\theta \subseteq [\![R]\!]$ we know $(\sigma,\sigma') \subseteq [\![R]\!]$, thus $(\sigma,\sigma') \models R$. Therefore, $\delta(C_0)\otimes\eta^{(State)} \overset{R}{\mapsto} \delta(C_0)\otimes\eta'^{(State)}$.

Lemma 275. *For all set A and $\mu_1,\mu_2 \in \mathbb{D}_A, p \in (0,1)$, $supp(\mu_1 \oplus_p \mu_2) = supp(\mu_1) \cup supp(\mu_2)$.*

Proof. For all set A and $\mu_1,\mu_2 \in \mathbb{D}_A, p \in (0,1)$, we have

$$\begin{aligned}
&supp(\mu_1 \oplus_p \mu_2) \\
&= \{a \mid (\mu_1 \oplus_p \mu_2)(a) > 0\} \\
&= \{a \mid p\cdot\mu_1(a) + (1-p)\cdot\mu_2(a) > 0\} \\
&= \{a \mid \mu_1(a) > 0 \vee \mu(a) > 0\} \\
&= \{a \mid \mu_1(a) > 0\} \cup \{a \mid \mu_2(a) > 0\} \\
&= supp(\mu_1) \cup supp(\mu_2).
\end{aligned}$$

Lemma 276. *For all set A and $\mu_1,\mu_2 \in \mathbb{D}_A, E \in A \to Prop$, if $\boldsymbol{Pr}_{a\sim\mu_1}[E(a)] > 0$, $\boldsymbol{Pr}_{a\sim\mu_1}[E(a)] > 0$ and $supp(\mu_1) \subseteq supp(\mu_2)$, then $supp(\mu_1|_E) \subseteq supp(\mu_2|_E)$.*

Proof. For all set A and $\mu_1,\mu_2 \in \mathbb{D}_A, E \in A \to \mathrm{Prop}$ such that $\mathbf{Pr}_{a\sim\mu_1}[E(a)] > 0$, $\mathbf{Pr}_{a\sim\mu_1}[E(a)] > 0$ and $supp(\mu_1) \subseteq supp(\mu_2)$, we have

$$\begin{aligned}
supp(\mu_1|_E) &= \{a \mid \mu_1|_E(a) > 0\} \\
&= \{a \mid \tfrac{\chi(E(a))\cdot\mu_1(a)}{\mathbf{Pr}_{a'\sim\mu_1}[E(a')]} > 0\} \\
&= \{a \mid E(a) \wedge \mu_1(a) > 0\} \\
&= \{a \mid E(a) \wedge a \in supp(\mu_1)\} \\
&\subseteq \{a \mid E(a) \wedge a \in supp(\mu_2)\} \\
&= \{a \mid E(a) \wedge \mu_2(a) > 0\} \\
&= \{a \mid \tfrac{\chi(E(a))\cdot\mu_2(a)}{\mathbf{Pr}_{a'\sim\mu_2}[E(a')]} > 0\} \\
&= \{a \mid \mu_2|_E(a) > 0\} \\
&= supp(\mu_2|_E).
\end{aligned}$$

Lemma 277. *For all η_1, η_2, if ${\eta_1}^{(Stmt)}(\textbf{\textit{skip}}) > 0$, ${\eta_2}^{(Stmt)}(\textbf{\textit{skip}}) > 0$ and $supp(\eta_1) \subseteq supp(\eta_2)$, then $supp(\eta_1|_{\textbf{\textit{skip}}}) \subseteq supp(\eta_2|_{\textbf{\textit{skip}}})$.*

Proof. For all η_1, η_2, if ${\eta_1}^{(Stmt)}(\textbf{skip}) > 0$, ${\eta_2}^{(Stmt)}(\textbf{skip}) > 0$ and $supp(\eta_1) \subseteq supp(\eta_2)$, by Lem. 2 and we know $\mathbf{Pr}_{(C,\sigma)\sim\eta_1}[C = \textbf{skip}] = \mathbf{Pr}_{C\sim{\eta_1}^{(State)}}[C = \textbf{skip}] = {\eta_1}^{(Stmt)}(\textbf{skip}) > 0$ and $\mathbf{Pr}_{(C,\sigma)\sim\eta_2}[C = \textbf{skip}] = \mathbf{Pr}_{C\sim{\eta_2}^{(State)}}[C = \textbf{skip}] = {\eta_2}^{(Stmt)}(\textbf{skip}) > 0$. From $supp(\eta_1) \subseteq supp(\eta_2)$ by Lem. 276 we have $supp(\eta_1|_{\lambda(C,\sigma).\ C=\textbf{skip}}) \subseteq supp(\eta_2|_{\lambda(C,\sigma).\ C=\textbf{skip}})$, i.e., $supp(\eta_1|_{\textbf{skip}}) = supp(\eta_2|_{\textbf{skip}})$.

Lemma 278. *For all set A and $\mu_1, \mu_2 \in \mathbb{D}_A, p \in [0,1]$, $\mu_1 \oplus_p \mu_2 = \mu_2 \oplus_{1-p} \mu_1$.*

Proof. For all set A and $\mu_1, \mu_2 \in \mathbb{D}_A, p \in [0,1]$, $\mu_1 \oplus_p \mu_2 = \lambda a.\ p \cdot \mu_1(a) + (1-p) \cdot \mu_2(a) = \lambda a.\ (1-p) \cdot \mu_2(a) + p \cdot \mu_1(a) = \mu_2 \oplus_{1-p} \mu_1$.

Lemma 279. *For all set A and $\mu_1, \mu_2, \mu_3 \in \mathbb{D}_A, p, p' \in (0,1)$, $(\mu_1 \oplus_p \mu_2) \oplus_{p'} \mu_3 = \mu_1 \oplus_{p\cdot p'} (\mu_2 \oplus_{\frac{p'(1-p)}{1-p\cdot p'}} \mu_3)$.*

Proof. For all set A and $\mu_1, \mu_2, \mu_3 \in \mathbb{D}_A, p, p' \in (0,1)$,

$$
\begin{aligned}
&(\mu_1 \oplus_p \mu_2) \oplus_{p'} \mu_3 \\
&= \lambda a.\ p' \cdot (\mu_1 \oplus_p \mu_2)(a) + (1-p')\mu_3(a) \\
&= \lambda a.\ p' \cdot (p \cdot \mu_1(a) + (1-p) \cdot \mu_2(a)) + (1-p')\mu_3(a) \\
&= \lambda a.\ p \cdot p' \cdot \mu_1(a) + p'(1-p) \cdot \mu_2(a) + (1-p')\mu_3(a) \\
&= \lambda a.\ p \cdot p' \cdot \mu_1(a) + (1 - p \cdot p') \cdot (\tfrac{p'(1-p)}{1-p\cdot p'} \cdot \mu_2(a) + \tfrac{1-p'}{1-p\cdot p'} \cdot \mu_3(a)) \\
&= \lambda a.\ p \cdot p' \cdot \mu_1(a) + (1 - p \cdot p') \cdot (\tfrac{p'(1-p)}{1-p\cdot p'} \cdot \mu_2(a) + (1 - \tfrac{p'(1-p)}{1-p\cdot p'}) \cdot \mu_3(a)) \\
&= \lambda a.\ p \cdot p' \cdot \mu_1(a) + (1 - p \cdot p') \cdot (\mu_2 \oplus_{\frac{p'(1-p)}{1-p\cdot p'}} \mu_3)(a) \\
&= \mu_1 \oplus_{p\cdot p'} (\mu_2 \oplus_{\frac{p'(1-p)}{1-p\cdot p'}} \mu_3).
\end{aligned}
$$

Lemma 280. *For all C and μ, if $\textbf{Nosplit}(C)$, then $\textbf{Nosplit}(\delta(C) \otimes \mu)$.*

Proof. For all C and μ such that $\textbf{Nosplit}(C)$, by Lem. 18 we know $supp(\delta(C) \otimes {\mu}^{(Stmt)}) = supp(\delta(C)) = \{C\}$. For all $C' \in supp(\delta(C) \otimes {\mu}^{(Stmt)})$, we have $C' = C$. From $\textbf{Nosplit}(C)$ we know $\textbf{Nosplit}(C')$.

Lemma 281. *For all η, θ, η', if $\textbf{Nosplit}(\eta)$ and $\eta \rightsquigarrow (\theta, \eta')$, then $\textbf{Nosplit}(\eta')$.*

Proof. For all η, θ, η' such that $\textbf{Nosplit}(\eta)$ and $\eta \rightsquigarrow (\theta, \eta')$, we have $\eta' = \lambda(C', \sigma').\ \sum_{C,\sigma}\{\eta(C,\sigma) \cdot p \mid (C,\sigma) \xrightarrow{p} (C',\sigma)\}$. For all $C' \in supp({\eta'}^{(Stmt)})$, by Lem. 22 we know $supp({\eta'}^{(Stmt)}) =$
$range(supp(\eta'))$, thus $C' \in range(supp(\eta'))$, so there exists σ' such that $(C', \sigma') \in supp(\eta')$, i.e., $\sum_{C,\sigma}\{\eta(C,\sigma) \cdot p \mid (C,\sigma) \xrightarrow{p} (C',\sigma)\} > 0$, thus there exists C and σ such that $\eta(C,\sigma) \cdot p > 0$ and $(C,\sigma) \xrightarrow{p} (C',\sigma)$. From $\eta(C,\sigma) > 0$ we know $(C,\sigma) \in supp(\eta)$, so $C \in range(supp(\eta))$. By Lem. 22 we know $range(supp(\eta)) = supp({\eta}^{(Stmt)})$, thus $C \in supp({\eta}^{(Stmt)})$. From $\textbf{Nosplit}(\eta)$ we know $\textbf{Nosplit}(C)$. From $(C,\sigma) \xrightarrow{p} (C',\sigma)$ by Lem. 51 we have $\textbf{Nosplit}(C')$.

Definition 86. $\mathbf{disablesplit}(Q, \eta)$ if and only if $\mathbf{disablesplit}(Q, C)$ for all $C \in supp(\eta^{(Stmt)})$.

Lemma 282. *For all* η *and* Q, *if* $\mathbf{Nosplit}(\eta)$, *then* $\mathbf{disablesplit}(Q, \eta)$.

Proof. For all η and Q such that $\mathbf{Nosplit}(\eta)$, for all $C \in supp(\eta^{(Stmt)})$, we have $\mathbf{Nosplit}(C)$, thus $\mathbf{disablesplit}(Q, C)$.

Lemma 283. *For all* η_1, η_2, p, *if* $\mathbf{Nosplit}(\eta_1)$ *and* $\mathbf{Nosplit}(\eta_2)$, *then* $\mathbf{Nosplit}(\eta_1 \oplus_p \eta_2)$.

Proof. For all η_1, η_2, p such that $\mathbf{Nosplit}(\eta_1)$ and $\mathbf{Nosplit}(\eta_2)$, there are three cases: $p = 0$, $p = 1$ or $0 < p < 1$. We prove the three cases respectively.

- $p = 0$.
  $\eta_1 \oplus_p \eta_2 = \eta_2$. From $\mathbf{Nosplit}(\eta_2)$ we know $\mathbf{Nosplit}(\eta_1 \oplus_p \eta_2)$.
- $p = 1$.
  $\eta_1 \oplus_p \eta_2 = \eta_1$. From $\mathbf{Nosplit}(\eta_1)$ we know $\mathbf{Nosplit}(\eta_1 \oplus_p \eta_2)$.
- $0 < p < 1$.
  By Lem. 11 we know ${\eta_1 \oplus_p \eta_2}^{(Stmt)} = {\eta_1}^{(Stmt)} \oplus_p {\eta_2}^{(Stmt)}$. From $0 < p < 1$ we know $supp((\eta_1 \oplus_p \eta_2)^{(Stmt)}) = supp({\eta_1}^{(Stmt)} \oplus_p {\eta_2}^{(Stmt)}) = supp({\eta_1}^{(Stmt)}) \cup supp({\eta_2}^{(Stmt)})$. For all $C \in supp((\eta_1 \oplus_p \eta_2)^{(Stmt)})$, we have $C \in supp({\eta_1}^{(Stmt)}) \cup supp({\eta_2}^{(Stmt)})$, thus $C \in supp({\eta_1}^{(Stmt)})$ or $C \in supp({\eta_2}^{(Stmt)})$. If $C \in supp({\eta_1}^{(Stmt)})$, from $\mathbf{Nosplit}(\eta_1)$ we know $\mathbf{Nosplit}(C)$. If $C \in supp({\eta_2}^{(Stmt)})$, from $\mathbf{Nosplit}(\eta_2)$ we know $\mathbf{Nosplit}(C)$.

Lemma 284 (Soundness of (SEQ-NST) rule). *For all* $C_1, C_2, R, G_1, G_2, I, P, M, Q$, *if* $R \vee G_2, G_1, I \models_{NST} \{P\}C_1\{M\}$, $R, G_2, true \models_{NST} \{M\}C_2\{Q\}$, $\mathbf{Nosplit}(C_2)$, $\mathbf{closed}(Q)$, $\mathbf{Id} \Rightarrow R$, $\mathbf{Id} \Rightarrow G_2$, $\mathbf{scl}(M)$ *and* $\forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x)$, *then* $R, G_1 \vee G_2, I \models_{NST} \{P\}C_1; C_2\{Q\}$.

Proof. For all $C_1, C_2, R, G_1, G_2, I, P, M, Q$ such that $R \vee G_2, G_1, I \models_{\text{NST}} \{P\}C_1\{M\}$, $R, G_2, \text{true} \models_{\text{NST}} \{M\}C_2\{Q\}$, $\mathbf{Nosplit}(C_2)$, $\mathbf{closed}(Q)$ and $\forall x \in fv(I).G_2 \Rightarrow \mathbf{Inv}(x)$, to prove $R, G_1 \vee G_2, I \models_{\text{NST}} \{P\}C_1; C_2\{Q\}$, we need to prove for all μ, if $\mu \models I \wedge P$, then $(init(C_1; C_2, \mu), R, I) \Longmapsto^n_{\text{NST}} \{P\}C_1; C_2\{Q\}$ for all n. For all μ such that $\mu \models I \wedge P$, from $R \vee G_2, G_1, I \models_{\text{NST}} \{P\}C_1\{M\}$ we know $(init(C_1, \mu), R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m. By Lem. 217 we know $init(C_1; C_2, \mu) = init(C_1, \mu); C_2$. For all n, to prove $(init(C_1; C_2, \mu), R, I) \Longmapsto^n_{\text{NST}} \{P\}C_1; C_2\{Q\}$, it suffices to prove for all η, if there exists η_1 such that $\eta = \eta_1; C_2$ and $(\eta_1, R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m, or there exists η_1, η_2, p such that $\eta = (\eta_1; C_2) \oplus_p \eta_2$, $0 < p < 1$, $(\eta_2, R, \text{true}) \Longmapsto^n_{\text{NST}} (G_2, Q)$, $\mathbf{Nosplit}(\eta_2)$ and $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}), R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m, or $(\eta, R, \text{true}) \Longmapsto^n_{\text{NST}} (G_2, Q)$, $\eta^{(State)} \models I$ and $\mathbf{Nosplit}(\eta)$, then $(\eta, R, I) \Longmapsto^n_{\text{NST}} (G_1 \vee G_2, Q)$. We prove it by induction on n.

- base case: $n = 0$. trivial.

– inductive case: $n = k + 1$.
IH: for all η, if there exists η_1 such that $\eta = \eta_1; C_2$ and $(\eta_1, R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m, or there exists η_1, η_2, p such that $\eta = (\eta_1; C_2) \oplus_p \eta_2$, $0 < p < 1$, $(\eta_2, R, \text{true}) \Longmapsto^n_{\text{NST}} (G_2, Q)$, $\mathbf{Nosplit}(\eta_2)$ and $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}), R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m, or $(\eta, R, \text{true}) \Longmapsto^k_{\text{NST}} (G_2, Q)$, $\eta^{(State)} \models I$ and $\mathbf{Nosplit}(\eta)$, then $(\eta, R, I) \Longmapsto^k_{\text{NST}} (G_1 \vee G_2, Q)$.
For all η such that there exists η_1 such that $\eta = \eta_1; C_2$ and $(\eta_1, R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m, or there exists η_1, η_2, p such that $\eta = (\eta_1; C_2) \oplus_p \eta_2$, $0 < p < 1$, $(\eta_2, R, \text{true}) \Longmapsto^{k+1}_{\text{NST}} (G_2, Q)$, $\mathbf{Nosplit}(\eta_2)$ and $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}), R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m, or $(\eta, R, \text{true}) \Longmapsto^{k+1}_{\text{NST}} (G_2, Q)$, $\eta^{(State)} \models I$ and $\mathbf{Nosplit}(\eta)$, we prove the three cases respectively.
- there exists η_1 such that $\eta = \eta_1; C_2$ and $(\eta_1, R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m.
  To prove $(\eta, R, I) \Longmapsto^{k+1}_{\text{NST}} (G_1 \vee G_2, Q)$, we need to prove
  * if ${\eta^{(Stmt)}}(\mathbf{skip}) > 0$, then ${\eta|_{\mathbf{skip}}}^{(State)} \models Q$.
    ${\eta^{(Stmt)}}(\mathbf{skip}) = {\eta_1; C_2}^{(Stmt)}(\mathbf{skip}) = \sum_\sigma \eta_1; C_2(\mathbf{skip}, \sigma) = 0$, which contradicts with ${\eta^{(Stmt)}}(\mathbf{skip}) > 0$.
  * $\eta^{(State)} \models I$.
    From $(\eta_1, R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m by Lem. 241 we know ${\eta_1}^{(State)} \models I$. By Lem. 201 we have ${\eta^{(State)}} = {\eta_1; C_2}^{(State)} = {\eta_1}^{(State)} \models I$.
  * for all η', if $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longmapsto^k_{\text{NST}} (G_1 \vee G_2, Q)$.
    For all η' such that $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, from $\eta = \eta_1; C_2$ we have $\eta_1; C_2 \overset{R}{\underset{I}{\rightarrowtail}} \eta'$.
    By Lem. 209 there exists η_1' such that $\eta' = \eta_1'; C_2$ and $\eta_1 \overset{R}{\underset{I}{\rightarrowtail}} \eta_1'$.
    From $(\eta_1, R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m by Lem. 241 we know $(\eta_1', R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m. From $\eta' = \eta_1'; C_2$ by IH we have $(\eta', R, I) \Longmapsto^k_{\text{NST}} (G_1 \vee G_2, Q)$.
  * for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G_1 \vee G_2]\!]$, ${\eta'}^{(State)} \models I$ and $(\eta', R, I) \Longmapsto^k_{\text{NST}} (G_1 \vee G_2, Q)$.
    For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, from $\eta = \eta_1; C_2$ we have $\eta_1; C_2 \hookrightarrow (\theta, \eta')$. There are three cases: ${\eta_1}^{(Stmt)}(\mathbf{skip}) = 1$, ${\eta_1}^{(Stmt)}(\mathbf{skip}) = 0$ or $0 < {\eta_1}^{(Stmt)}(\mathbf{skip}) < 1$. We prove the three cases respectively.
    · ${\eta_1}^{(Stmt)}(\mathbf{skip}) = 1$.
      By Lem. 25 we know ${\eta_1}^{(Stmt)} = \delta(\mathbf{skip})$. From $\eta_1; C_2 \hookrightarrow (\theta, \eta')$ by Lem. 210 and Lem. 193 we know $\theta = \{(\sigma, \sigma) \mid \sigma \in supp({\eta_1}^{(State)})\} \subseteq [\![\mathbf{Id}]\!] \subseteq [\![G_1 \vee G_2]\!]$. and $\eta' = \delta(C_2) \otimes {\eta_1}^{(State)}$. By Lem. 19 we know ${\eta'}^{(State)} = {\eta_1}^{(State)} \models I$. From $(\eta_1, R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m and ${\eta^{(Stmt)}}(\mathbf{skip}) = 1 > 0$ by Lem. 241 we know ${\eta_1|_{\mathbf{skip}}}^{(State)} \models M$. From ${\eta^{(Stmt)}}(\mathbf{skip}) = 1$ by Lem. 199 we know $\eta_1|_{\mathbf{skip}} = \eta_1$, thus ${\eta_1}^{(State)} = {\eta_1|_{\mathbf{skip}}}^{(State)} \models M$. From $R, G_2, \text{true} \models_{\text{NST}} \{M\} C_2 \{Q\}$ and $\eta' = \delta(C_2) \otimes {\eta_1}^{(State)} = init(C_2, {\eta_1}^{(State)})$ we have $(\eta', R, \text{true}) \Longmapsto^k_{\text{NST}} (G_2, Q)$. From $\mathbf{Nosplit}(C_2)$ by Lem. 280 we know $\mathbf{Nosplit}(\delta(C_2) \otimes {\eta_1}^{(State)})$, i.e., $\mathbf{Nosplit}(\eta_2')$.

From $(\eta', R, \text{true}) \Longrightarrow^{k}_{\text{NST}} (G_2, Q)$, ${\eta'}^{(State)} \models I$ and $\mathbf{Nosplit}(\eta'_2)$ by IH we have $(\eta', R, I) \Longrightarrow^{k}_{\text{NST}} (G_1 \vee G_2, Q)$.

- ${\eta_1}^{(Stmt)}(\mathbf{skip}) = 0$.
  From $\eta_1; C_2 \hookrightarrow (\theta, \eta')$ by Lem. 215 there exists η'_1 such that $\eta' = \eta'_1; C_2$ and $\eta_1 \hookrightarrow (\theta, \eta'_1)$. From $(\eta_1, R \vee G_2, I) \Longrightarrow^{m}_{\text{NST}} (G_1, M)$ for all m by Lem. 241 we have $\theta \subseteq [\![G_1]\!] \subseteq [\![G_1 \vee G_2]\!]$, ${\eta'_1}^{(State)} \models I$ and $(\eta'_1, R \vee G_2, I) \Longrightarrow^{m}_{\text{NST}} (G_1, M)$ for all m. By Lem. 201 we know ${\eta'}^{(State)} = {\eta'_1; C_2}^{(State)} = {\eta'_1}^{(State)} \models I$. From $\eta' = \eta'_1; C_2$ and $(\eta'_1, R \vee G_2, I) \Longrightarrow^{m}_{\text{NST}} (G_1, M)$ for all m by IH we have $(\eta', R, I) \Longrightarrow^{k}_{\text{NST}} (G_1 \vee G_2, Q)$.
- $0 < {\eta_1}^{(Stmt)}(\mathbf{skip}) < 1$.
  Let $p \stackrel{\text{def}}{=} {\eta_1}^{(Stmt)}(\mathbf{skip})$, then $0 < p < 1$. By Lem. 247 there exists η_{11} and η_{12} such that $\eta_1 = \eta_{11} \oplus_p \eta_{12}$, ${\eta_{11}}^{(Stmt)}(\mathbf{skip}) = 1$ and ${\eta_{12}}^{(Stmt)}(\mathbf{skip}) = 0$. From ${\eta_{11}}^{(Stmt)}(\mathbf{skip}) = 1$ by Lem. 25 we have ${\eta_{11}}^{(Stmt)} = \delta(\mathbf{skip})$. By Lem. 190 we know $\mathit{nextsplit}(\eta_{11}) = \{\mathit{nextsplit}(\mathbf{skip})\} = \{\mathbf{split}(\text{true})\}$. From $0 < p < 1$ by Lem. 243 we know $\mathit{nextsplit}(\eta_1)$
  $= \mathit{nextsplit}(\eta_{11} \oplus_p \eta_{12}) = \mathit{nextsplit}(\eta_{11}) \cup \mathit{nextsplit}(\eta_{12}) \supseteq \mathit{nextsplit}(\eta_{11}) = \{\mathbf{split}(\text{true})\}$. By Lem. 214 we know $\mathit{nextsplit}(\eta_1; C_2) = \mathit{nextsplit}(\eta_1) \supseteq \{\mathbf{split}(\text{true})\}$. From $\eta_1; C_2 \hookrightarrow (\theta, \eta')$ by Lem. 191 we have $\eta_1; C_2 \rightsquigarrow (\theta, \eta')$. By Lem. 248 we know $\eta_1; C_2 = (\eta_{11} \oplus_p \eta_{12}); C_2 = (\eta_{11}; C_2) \oplus_p (\eta_{12}; C_2)$, thus $(\eta_{11}; C_2) \oplus_p (\eta_{12}; C_2) \rightsquigarrow (\theta, \eta')$. From $0 < p < 1$ by Lem. 245 there exists $\theta_1, \theta_2, \eta'_1, \eta'_2$ such that $\eta' = \eta'_1 \oplus_p \eta'_2$, $\theta = \theta_1 \cup \theta_2$, $\eta_{11}; C_2 \rightsquigarrow (\theta_1, \eta'_1)$ and $\eta_{12}; C_2 \rightsquigarrow (\theta_2, \eta'_2)$. From ${\eta_{11}}^{(Stmt)}(\mathbf{skip}) = 1$ by Lem. 13 we know $\eta_{11} = \delta(\mathbf{skip}) \otimes {\eta_{11}}^{(State)}$. By Lem. 19 we have ${\eta_{11}}^{(Stmt)} = \delta(\mathbf{skip})$. From $\eta_{11}; C_2 \hookrightarrow (\theta_1, \eta'_1)$ by Lem. 210 and Lem. 193 we know $\theta_1 = \{(\sigma, \sigma) \mid \sigma \in \mathit{supp}({\eta_{11}}^{(State)})\} \subseteq [\![\mathbf{Id}]\!] \subseteq [\![G_1 \vee G_2]\!]$. and $\eta' = \delta(C_2) \otimes {\eta_{11}}^{(State)}$. From $\eta_{12}; C_2 \hookrightarrow (\theta_2, \eta'_2)$ by Lem. 215 there exists η'_{21} such that $\eta'_2 = \eta'_{21}; C_2$ and $\eta_{12} \hookrightarrow (\theta_2, \eta'_{21})$. From ${\eta_{11}}^{(Stmt)} = \delta(\mathbf{skip})$ by Lem. 192 we know $\eta_{11} \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}({\eta_{11}}^{(State)})\}, \delta(\mathbf{skip}) \otimes {\eta_{11}}^{(State)})$, i.e., $\eta_{11} \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}({\eta_{11}}^{(State)})\}, \eta_{11})$. From $\eta_{12} \hookrightarrow (\theta_2, \eta'_{21})$ and $0 < p < 1$ by Lem. 246 we know $\eta_{11} \oplus_p \eta_{12} \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}({\eta_{11}}^{(State)})\} \cup \theta_2, \eta_{11} \oplus_p \eta'_{21})$, i.e., $\eta_1 \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}({\eta_{11}}^{(State)})\} \cup \theta_2, \eta_{11} \oplus_p \eta'_{21})$. From $\mathit{nextsplit}(\eta_1) \supseteq \{\mathbf{split}(\text{true})\}$ by Lem. 191 we know $\eta_1 \hookrightarrow (\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}({\eta_{11}}^{(State)})\} \cup \theta_2, \eta_{11} \oplus_p \eta'_{21})$. From $(\eta_1, R \vee G_2, I) \Longrightarrow^{m}_{\text{NST}} (G_1, M)$ for all m by Lem. 241 we have $\{(\sigma, \sigma) \mid \sigma \in \mathit{supp}({\eta_{11}}^{(State)})\} \cup \theta_2 \subseteq [\![G_1]\!]$, $(\eta_{11} \oplus_p \eta'_{21})^{(State)} \models I$ and $(\eta_{11} \oplus_p \eta'_{21}, R \vee G_2, I) \Longrightarrow^{m}_{\text{NST}} (G_1, M)$ for all m, thus $\theta_2 \subseteq [\![G_1]\!] \subseteq [\![G_1 \vee G_2]\!]$. From $\theta_1 \subseteq [\![G_1 \vee G_2]\!]$ we have $\theta = \theta_1 \cup \theta_2 \subseteq [\![G_1 \vee G_2]\!]$. From $\eta' = \eta'_1 \oplus_p \eta'_2 = (\delta(C_2) \otimes {\eta_{11}}^{(State)}) \oplus_p (\eta'_{21}; C_2)$ by Lem. 12, Lem. 19 and Lem. 201 we know ${\eta'}^{(State)} = {(\delta(C_2) \otimes {\eta_{11}}^{(State)})}^{(State)} \oplus_p {(\eta'_{21}; C_2)}^{(State)} = {\eta_{11}}^{(State)} \oplus_p {\eta'_{21}}^{(State)} = (\eta_{11} \oplus_p \eta'_{21})^{(State)}$. From $(\eta_{11} \oplus_p \eta'_{21})^{(State)} \models$

I we have ${\eta'}^{(State)} \models I$. From $\eta' = (\delta(C_2) \otimes {\eta_{11}}^{(State)}) \oplus_p (\eta'_{21}; C_2)$ by Lem. 278 we know $\eta' = (\eta'_{21}; C_2) \oplus_{1-p} (\delta(C_2) \otimes {\eta_{11}}^{(State)})$. From $0 < p < 1$ we have $0 < 1 - p < 1$. From $(\eta_1, R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m and ${\eta_1}^{(Stmt)}(\mathbf{skip}) > 0$ by Lem. 241 we have ${\eta_1|_{\mathbf{skip}}}^{(State)} \models M$. From $\eta_1 = \eta_{11} \oplus_p \eta_{12}$ by Lem. 278 we know $\eta_1 = \eta_{12} \oplus_{1-p} \eta_{11}$. From ${\eta_{12}}^{(Stmt)}(\mathbf{skip}) = 0$ and $(\eta_{12} \oplus_{1-p} \eta_{11})^{(Stmt)}(\mathbf{skip}) = {\eta_1}^{(Stmt)}(\mathbf{skip}) > 0$ by Lem. 234 we have $\eta_1|_{\mathbf{skip}} = (\eta_{12} \oplus_{1-p} \eta_{11})|_{\mathbf{skip}} = \eta_{11}|_{\mathbf{skip}}$. From ${\eta_{11}}^{(Stmt)}(\mathbf{skip}) = 1$ by Lem. 199 we know $\eta_{11}|_{\mathbf{skip}} = \eta_{11}$, thus $\eta_1|_{\mathbf{skip}} = \eta_{11}|_{\mathbf{skip}} = \eta_{11}$. From ${\eta_1|_{\mathbf{skip}}}^{(State)} \models M$ we have ${\eta_{11}}^{(State)} \models M$. From $R, G_2, \text{true} \models_{\text{NST}} \{M\} C_2 \{Q\}$ we know $(\delta(C_2) \otimes {\eta_{11}}^{(State)}, R, \text{true}) \Longmapsto^k_{\text{NST}} (G_2, Q)$. From $\mathbf{Nosplit}(C_2)$ by Lem. 280 we have $\mathbf{Nosplit}(\delta(C_2) \otimes {\eta_{11}}^{(State)})$. By Lem. 278 we know $\eta_{11} \oplus_p \eta'_{21} = \eta'_{21} \oplus_{1-p} \eta_{11}$. From $(\eta_{11} \oplus_p \eta'_{21}, R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m and $\eta_{11} = \delta(\mathbf{skip}) \otimes {\eta_{11}}^{(State)}$ we know $(\eta'_{21} \oplus_{1-p} (\delta(\mathbf{skip}) \otimes {\eta_{11}}^{(State)}), R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m. From $\eta' = (\delta(C_2) \otimes {\eta_{11}}^{(State)}) \oplus_p (\eta'_{21}; C_2)$, $0 < 1 - p < 1$, $(\delta(C_2) \otimes {\eta_{11}}^{(State)}, R, \text{true}) \Longmapsto^k_{\text{NST}} (G_2, Q)$, $\mathbf{Nosplit}(\delta(C_2) \otimes {\eta_{11}}^{(State)})$ and $(\eta'_{21} \oplus_{1-p} (\delta(\mathbf{skip}) \otimes {\eta_{11}}^{(State)}), R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m by IH we have $(\eta', R, I) \Longmapsto^k_{\text{NST}} (G_1 \vee G_2, Q)$.

- there exists η_1, η_2, p such that $\eta = (\eta_1; C_2) \oplus_p \eta_2$, $0 < p < 1$, $(\eta_2, R, \text{true}) \Longmapsto^{k+1}_{\text{NST}} (G_2, Q)$, $\mathbf{Nosplit}(\eta_2)$ and $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}), R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m, to prove $(\eta, R, I) \Longmapsto^{k+1}_{\text{NST}} (G_1 \vee G_2, Q)$, we need to prove
  - if ${\eta}^{(Stmt)}(\mathbf{skip}) > 0$, then ${\eta|_{\mathbf{skip}}}^{(State)} \models Q$.
    ${\eta_1; C_2}^{(Stmt)}(\mathbf{skip}) = \sum_\sigma {\eta_1; C_2}(\mathbf{skip}, \sigma) = 0$, thus ${\eta}^{(Stmt)}(\mathbf{skip}) = {(\eta_1; C_2 \oplus_p \eta_2)}^{(Stmt)}(\mathbf{skip}) = p \cdot {\eta_1; C_2}^{(Stmt)}(\mathbf{skip}) + (1-p) \cdot {\eta_2}^{(Stmt)}(\mathbf{skip}) = (1-p) \cdot {\eta_2}^{(Stmt)}(\mathbf{skip})$. From ${\eta}^{(Stmt)}(\mathbf{skip}) > 0$ we know ${\eta_2}^{(Stmt)}(\mathbf{skip}) > 0$. From $(\eta_2, R, \text{true}) \Longmapsto^{k+1}_{\text{NST}} (G_2, Q)$ we know ${\eta_2|_{\mathbf{skip}}}^{(State)} \models Q$. From ${\eta_1; C_2}^{(Stmt)}(\mathbf{skip}) = 0$ by Lem. 234 we know $\eta|_{\mathbf{skip}} = (\eta_1; C_2 \oplus_p \eta_2)|_{\mathbf{skip}} = \eta_2|_{\mathbf{skip}}$, thus ${\eta|_{\mathbf{skip}}}^{(State)} \models Q$.
  - ${\eta}^{(State)} \models I$.
    From $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}), R \vee G_2, I) \Longmapsto^m_{\text{NST}} (G_1, M)$ for all m we know
    ${(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}))}^{(State)} \models I$. From

$$
\begin{aligned}
& {(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}))}^{(State)} \\
&= {\eta_1}^{(State)} \oplus_p {(\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})}^{(State)} && \text{(by Lem. 12)} \\
&= {\eta_1}^{(State)} \oplus_p {\eta_2}^{(State)} && \text{(by Lem. 19)} \\
&= {\eta_1; C_2}^{(State)} \oplus_p {\eta_2}^{(State)} && \text{(by Lem. 201)} \\
&= {((\eta_1; C_2) \oplus_p \eta_2)}^{(State)} && \text{(by Lem. 12)} \\
&= {\eta}^{(State)}
\end{aligned}
$$

    we know ${\eta}^{(State)} \models I$.

* for all η', if $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longmapsto^{k}_{\text{NST}} (G_1 \vee G_2, Q)$.

  For all η' such that $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, there exists η'' and b such that $\eta \overset{R}{\mapsto} \eta''$, $\eta''|_b = \eta'$ and $\eta'^{(State)} \models I$. From $(\eta_1; C_2) \oplus_p \eta_2 \overset{R}{\mapsto} \eta'$ and $0 < p < 1$ we know there exists η_1'', η_2'', p'' such that $\eta_1; C_2 \overset{R}{\mapsto} \eta_1''$, $\eta_2 \overset{R}{\mapsto} \eta_2''$, $0 < p'' < 1$ and $\eta'' = \eta_1'' \oplus_{p''} \eta_2''$. From $\eta_1; C_2 \overset{R}{\mapsto} \eta_1''$ by Lem. 203 there exists η_{11}'' such that $\eta_1'' = \eta_{11}''; C_2$ and $\eta_1 \overset{R}{\mapsto} \eta_{11}''$, thus $\eta' = \eta''|_b = (\eta_1'' \oplus_{p''} \eta_2'')|_b = (\eta_{11}''; C_2 \oplus_{p''} \eta_2'')|_b$. Let $p_1 \overset{\text{def}}{=} [\![\mathbf{Pr}(b)]\!]_{\eta_{11}''^{(State)}}$ and $p_2 \overset{\text{def}}{=} [\![\mathbf{Pr}(b)]\!]_{\eta_2''^{(State)}}$, by Lem. 201 we know $\eta_1''; C_2{}^{(State)} = \eta_1''^{(State)}$, thus $[\![\mathbf{Pr}(b)]\!]_{\eta_{11}''; C_2{}^{(State)}} = [\![\mathbf{Pr}(b)]\!]_{\eta_{11}''^{(State)}} = p_1$. By Lem. 237 we know $[\![\mathbf{Pr}(b)]\!]_{\eta''^{(State)}} = [\![\mathbf{Pr}(b)]\!]_{\eta_1'' \oplus_{p''} \eta_2''^{(State)}} = p'' \cdot [\![\mathbf{Pr}(b)]\!]_{\eta_1''^{(State)}} + (1 - p'') \cdot [\![\mathbf{Pr}(b)]\!]_{\eta_2''^{(State)}} = p'' \cdot p_1 + (1 - p') \cdot p_2$. From $\eta''|_b = \eta'$ by Lem. 205 we know $[\![\mathbf{Pr}(b)]\!]_{\eta''^{(State)}} > 0$, thus $p'' \cdot p_1 + (1 - p') \cdot p_2 > 0$. There are three cases: $p_1 = 0 \wedge p_2 > 0$, $p_1 > 0 \wedge p_2 = 0$, or $p_1 > 0 \wedge p_2 > 0$. We prove the three cases respectively.

  · $p_1 = 0 \wedge p_2 > 0$.
  From $[\![\mathbf{Pr}(b)]\!]_{\eta_{11}''; C_2{}^{(State)}} = p_1$, $[\![\mathbf{Pr}(b)]\!]_{\eta_2''^{(State)}} = p_2$ and $0 < p'' < 1$ by Lem. 238 we know $\eta' = (\eta_1''; C_2 \oplus_{p''} \eta_2'')|_b = \eta_2''|_b$. From $\eta_2 \overset{R}{\mapsto} \eta_2''$ we know $\eta_2 \overset{R}{\underset{\text{true}}{\rightarrowtail}} \eta'$. From $(\eta_2, R, \text{true}) \Longmapsto^{k+1}_{\text{NST}} (G_2, Q)$ we have $(\eta', R, \text{true}) \Longmapsto^{k}_{\text{NST}} (G_2, Q)$. From $\eta_2 \overset{R}{\underset{\text{true}}{\rightarrowtail}} \eta'$ by Lem. 188 we know $supp(\eta'^{(Stmt)}) \subseteq supp(\eta_2{}^{(Stmt)})$. From $\mathbf{Nosplit}(\eta_2)$ by Lem. 239 we know $\mathbf{Nosplit}(\eta')$. From $(\eta', R, \text{true}) \Longmapsto^{k}_{\text{NST}} (G_2, Q), \eta'^{(Stmt)} \models I$ and $\mathbf{Nosplit}(\eta')$ by IH we have $(\eta', R, I) \Longmapsto^{k}_{\text{NST}} (G_1 \vee G_2, Q)$.

  · $p_1 > 0 \wedge p_2 = 0$.
  From $[\![\mathbf{Pr}(b)]\!]_{\eta_{11}''; C_2{}^{(State)}} = p_1$, $[\![\mathbf{Pr}(b)]\!]_{\eta_2''^{(State)}} = p_2$ and $0 < p'' < 1$ by Lem. 238 we know $\eta' = (\eta_1''; C_2 \oplus_{p''} \eta_2'')|_b = \eta_1''; C_2|_b$. From $[\![\mathbf{Pr}(b)]\!]_{\eta_{11}''^{(State)}} = p_1 > 0$ by Lem. 208 we know $\eta_1''; C_2|_b = \eta_1''|_b; C_2$, thus $\eta' = \eta_1''|_b; C_2$. From $\eta_2 \overset{R}{\mapsto} \eta_2''$ by Lem. 178 we know $\eta_2{}^{(State)} \overset{R}{\mapsto} \eta_2''^{(State)}$. By Lem. 240 we know $\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)} \overset{R}{\mapsto} \delta(\mathbf{skip}) \otimes \eta_2''^{(State)}$. From $\eta_1 \overset{R}{\mapsto} \eta_{11}''$, $0 < p < 1$ and $0 < p'' < 1$ by Lem. 236 we know $\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}) \overset{R}{\mapsto} \eta_{11}'' \oplus_{p''} (\delta(\mathbf{skip}) \otimes \eta_2''^{(State)})$. By Lem. 19 we know $(\delta(\mathbf{skip}) \otimes \eta_2''^{(State)})^{(State)} = \eta_2''^{(State)}$, thus $[\![\mathbf{Pr}(b)]\!]_{(\delta(\mathbf{skip}) \otimes \eta_2''^{(State)})^{(State)}} = [\![\mathbf{Pr}(b)]\!]_{\eta_2''^{(State)}} = p_2 = 0$. From $[\![\mathbf{Pr}(b)]\!]_{\eta_{11}''^{(State)}} = p_1 > 0$ and $0 < p'' < 1$ by Lem. 238 we know $(\eta_{11}'' \oplus_{p''} (\delta(\mathbf{skip}) \otimes \eta_2''^{(State)}))|_b = \eta_{11}''|_b$. By

Lem. 201 we know ${\eta''_{11}|_b}^{(State)} = ({\eta''_{11}|_b; C_2})^{(State)} = {\eta'}^{(State)} \models I$. From $\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}) \overset{R}{\mapsto} \eta''_{11} \oplus_{p''} (\delta(\mathbf{skip}) \otimes {\eta''_2}^{(State)})$, $(\eta''_{11} \oplus_{p''} (\delta(\mathbf{skip}) \otimes {\eta''_2}^{(State)}))|_b = \eta''_{11}|_b$ and ${\eta''_{11}|_b}^{(State)} \models I$ we know $\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}) \underset{I}{\overset{R}{\rightarrowtail}} \eta''_{11}|_b$. From $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}), R \vee G_2, I) \Longrightarrow^m_{\text{NST}} (G_1, M)$ for all m by Lem. 241 we know $(\eta''_{11}|_b, R \vee G_2, I) \Longrightarrow^m_{\text{NST}} (G_1, M)$ for all m. From $\eta' = \eta''_{11}|_b; C_2$ by IH we have $(\eta', R, I) \Longrightarrow^k_{\text{NST}} (G_1 \vee G_2, Q)$.

- $p_1 > 0 \wedge p_2 > 0$.

  Let $p' \overset{\text{def}}{=} \frac{p'' \cdot p_1}{p'' \cdot p_1 + (1-p'') \cdot p_2}$, then $0 < p' < 1$. From $[\![\mathbf{Pr}(b)]\!]_{{\eta''_{11}; C_2}^{(State)}} = p_1$, $[\![\mathbf{Pr}(b)]\!]_{{\eta''_2}^{(State)}} = p_2$ and $0 < p'' < 1$ by Lem. 238 we know $\eta' = (\eta''_{11}; C_2 \oplus_{p''} \eta''_2)|_b = (\eta''_{11}; C_2)|_b \oplus_{p'} \eta''_2|_b$. By Lem. 208 we know $(\eta''_{11}; C_2)|_b = \eta''_{11}|_b; C$, thus $\eta' = (\eta''_{11}; C_2)|_b \oplus_{p'} \eta''_2|_b = \eta''_{11}|_b; C_2 \oplus_{p'} \eta''_2|_b$. From $\eta_2 \overset{R}{\mapsto} \eta''_2$ by Lem. 178 we know ${\eta_2}^{(State)} \overset{R}{\mapsto} {\eta''_2}^{(State)}$. By Lem. 240 we know $\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)} \overset{R}{\mapsto} \delta(\mathbf{skip}) \otimes {\eta''_2}^{(State)}$. From $\eta_1 \overset{R}{\mapsto} \eta''_{11}$, $0 < p < 1$ and $0 < p'' < 1$ by Lem. 236 we know $\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}) \overset{R}{\mapsto} \eta''_{11} \oplus_{p''} (\delta(\mathbf{skip}) \otimes {\eta''_2}^{(State)})$. By Lem. 19 we know $(\delta(\mathbf{skip}) \otimes {\eta''_2}^{(State)})^{(State)} = {\eta''_2}^{(State)}$, thus $[\![\mathbf{Pr}(b)]\!]_{(\delta(\mathbf{skip}) \otimes {\eta''_2}^{(State)})^{(State)}} = [\![\mathbf{Pr}(b)]\!]_{{\eta''_2}^{(State)}} = p_2 > 0$. From $[\![\mathbf{Pr}(b)]\!]_{{\eta''_{11}}^{(State)}} = p_1 > 0$ and $0 < p'' < 1$ by Lem. 238 we know $(\eta''_{11} \oplus_{p''} (\delta(\mathbf{skip}) \otimes {\eta''_2}^{(State)}))|_b = \eta''_{11}|_b \oplus_{p'} (\delta(\mathbf{skip}) \otimes {\eta''_2}^{(State)})|_b$. From $[\![\mathbf{Pr}(b)]\!]_{{\eta''_2}^{(State)}} = p_2 > 0$ by Lem. 242 and Lem. 206 we know $(\delta(\mathbf{skip}) \otimes {\eta''_2}^{(State)})|_b = \delta(\mathbf{skip}) \otimes {\eta''_2}^{(State)}|_b = \delta(\mathbf{skip}) \otimes {\eta''_2|_b}^{(State)}$, thus $(\eta''_{11} \oplus_{p''} (\delta(\mathbf{skip}) \otimes {\eta''_2}^{(State)}))|_b = \eta''_{11}|_b \oplus_{p'} (\delta(\mathbf{skip}) \otimes {\eta''_2|_b}^{(State)})$. By Lem. 19 and Lem. 201 we know $(\eta''_{11}|_b \oplus_{p'} (\delta(\mathbf{skip}) \otimes {\eta''_2|_b}^{(State)}))^{(State)} = {\eta''_{11}|_b}^{(State)} \oplus_{p'} (\delta(\mathbf{skip}) \otimes {\eta''_2|_b}^{(State)})^{(State)} = ({\eta''_{11}|_b; C_2})^{(State)} \oplus_{p'} {\eta''_2|_b}^{(State)} = (\eta''_{11}|_b; C_2 \oplus_{p'} \eta''_2|_b)^{(State)} = {\eta'}^{(State)} \models I$. From $\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}) \overset{R}{\mapsto} \eta''_{11} \oplus_{p''} (\delta(\mathbf{skip}) \otimes {\eta''_2}^{(State)})$, $(\eta''_{11} \oplus_{p''} (\delta(\mathbf{skip}) \otimes {\eta''_2}^{(State)}))|_b = \eta''_{11}|_b \oplus_{p'} (\delta(\mathbf{skip}) \otimes {\eta''_2|_b}^{(State)})$ and $(\eta''_{11}|_b \oplus_{p'} (\delta(\mathbf{skip}) \otimes {\eta''_2|_b}^{(State)}))^{(State)} \models I$ we know $\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}) \underset{I}{\overset{R}{\rightarrowtail}} \eta''_{11}|_b \oplus_{p'} (\delta(\mathbf{skip}) \otimes {\eta''_2|_b}^{(State)})$. From $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}), R \vee G_2, I) \Longrightarrow^m_{\text{NST}} (G_1, M)$ for all m by Lem. 241 we know $(\eta''_{11}|_b \oplus_{p'} (\delta(\mathbf{skip}) \otimes {\eta''_2|_b}^{(State)}), R \vee G_2, I) \Longrightarrow^m_{\text{NST}} (G_1, M)$

for all m. From $\eta_2 \overset{R}{\mapsto} \eta_2''$ we know $\eta_2 \overset{R}{\underset{\text{true}}{\rightarrowtail}} \eta_2''|_b$. From $(\eta_2, R, \text{true}) \Longmapsto^{k+1}_{\text{NST}}$ (G_2, Q) we know $(\eta_2''|_b, R, \text{true}) \Longmapsto^{k}_{\text{NST}} (G_2, Q)$. From $\eta_2 \overset{R}{\underset{\text{true}}{\rightarrowtail}} \eta_2''|_b$ by Lem. 188 we know $supp(\eta_2''|_b{}^{(Stmt)}) \subseteq supp(\eta_2{}^{(Stmt)})$. From $\mathbf{Nosplit}(\eta_2)$ by Lem. 239 we know $\mathbf{Nosplit}(\eta_2''|_b)$. From $\eta' = \eta_{11}''|_b; C_2 \oplus_{p'} \eta_2''|_b$, $(\eta_2''|_b, R, \text{true}) \Longmapsto^{k}_{\text{NST}}$ (G_2, Q), $\mathbf{Nosplit}(\eta_2''|_b)$ and $(\eta_{11}''|_b \oplus_{p'} (\delta(\mathbf{skip}) \otimes \eta_2''|_b{}^{(State)}), R \vee G_2, I) \Longmapsto^{m}_{\text{NST}} (G_1, M)$ for all m by IH we have $(\eta', R, I) \Longmapsto^{k}_{\text{NST}}$ $(G_1 \vee G_2, Q)$.

* for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G_1 \vee G_2]\!]$, $\eta'^{(State)} \models I$ and $(\eta', R, I) \Longmapsto^{k}_{\text{NST}} (G_1 \vee G_2, Q)$.
  For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, by Lem. 243 and Lem. 214 we know we have $nextsplit(\eta) = nextsplit(\eta_1; C_2 \oplus \eta_2) = nextsplit(\eta_1; C_2) \cup nextsplit(\eta_2) = nextsplit(\eta_1) \cup nextsplit(\eta_2) \supseteq nextsplit(\eta_2)$. From $\mathbf{Nosplit}(\eta_2)$ by Lem. 244 we know $nextsplit(\eta_2) = \{\mathbf{split}(\text{true})\}$, thus $nextsplit(\eta) \supseteq nextsplit(\eta_2) = \{\mathbf{split}(\text{true})\}$. From $\eta \hookrightarrow (\theta, \eta')$ by Lem. 191 we know $\eta \rightsquigarrow (\theta, \eta')$, thus $\eta_1; C_2 \oplus_p \eta_2 \rightsquigarrow (\theta, \eta')$, by Lem. 245 we know there exists $\eta_1', \eta_2', \theta_1, \theta_2$ such that $\eta' = \eta_1' \oplus_p \eta_2'$, $\theta = \theta_1 \cup \theta_2$, $\eta_1; C_2 \rightsquigarrow (\theta_1, \eta_1')$ and $\eta_2 \rightsquigarrow (\theta_2, \eta_2')$. There are three cases: $\eta_1{}^{(Stmt)}(\mathbf{skip}) = 1$, $\eta_1{}^{(Stmt)}(\mathbf{skip}) = 0$, or $0 < \eta_1{}^{(Stmt)}(\mathbf{skip}) < 1$. We prove the three cases respectively.

  · $\eta_1{}^{(Stmt)}(\mathbf{skip}) = 1$.
    By Lem. 13 we know $\eta_1 = \delta(\mathbf{skip}) \otimes \eta_1{}^{(State)}$. By Lem. 18 we know $\eta_1{}^{(Stmt)} = \delta(\mathbf{skip})$. From $\eta_1; C_2 \rightsquigarrow (\theta_1, \eta_1')$ by Lem. 210 and Lem. 193 we know $\theta_1 = \{(\sigma, \sigma) \mid \sigma \in supp(\eta_1{}^{(State)})\} \subseteq [\![\mathbf{Id}]\!]$ and $\eta_1' = \delta(C_2) \otimes \eta_1{}^{(State)}$. By Lem. 192 we know $\eta_1 \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in supp(\eta_1{}^{(State)})\}, \delta(\mathbf{skip}) \otimes \eta_1{}^{(State)})$, i.e., $\eta_1 \rightsquigarrow (\theta_1, \eta_1)$. From $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}), R \vee G_2, I) \Longmapsto^{m}_{\text{NST}} (G_1, M)$ for all m by Lem. 241 we know $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}))^{(State)} \models I$. By Lem. 12 and Lem. 19 we know $\eta_1{}^{(State)} \oplus_p \eta_2{}^{(State)} = (\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}))^{(State)} \models I$. From $\mathbf{Nosplit}(\eta_2)$ by Lem. 244 we know $nextsplit(\eta_2) = \{\mathbf{split}(\text{true})\}$. From $\eta_2 \rightsquigarrow (\theta_2, \eta_2')$ by Lem. 191 we have $\eta_2 \hookrightarrow (\theta_2, \eta_2')$. From $(\eta_2, R, \text{true}) \Longmapsto^{k+1}_{\text{NST}} (G_2, Q)$ we know $\theta_2 \subseteq [\![G_2]\!]$ and $(\eta_2', R, \text{true}) \Longmapsto^{k}_{\text{NST}}$ (G_2, Q), thus $\theta = \theta_1 \cup \theta_2 \subseteq [\![\mathbf{Id}]\!] \cup [\![G_2]\!] = [\![G_2]\!] \subseteq [\![G_1 \vee G_2]\!]$. From $\theta_2 \subseteq [\![G_2]\!]$ and $\forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x)$ we know $\forall x \in fv(I), (\sigma, \sigma') \in \theta.\ \sigma'(x) = \sigma(x)$. From $\eta_2 \rightsquigarrow (\theta_2, \eta_2')$ by Lem. 249 we know $\eta_2'^{(State)}|_{fv(I)} = \eta_2{}^{(State)}|_{fv(I)}$. From $\eta' = \eta_1' \oplus_p \eta_2' = (\delta(C_2) \otimes \eta_1{}^{(State)}) \oplus_p \eta_2'$ by Lem. 12 and Lem. 19 we know $\eta'^{(State)} = (\delta(C_2) \otimes \eta_1{}^{(State)})^{(State)} \oplus_p \eta_2'{}^{(State)} = \eta_1{}^{(State)} \oplus_p \eta_2'{}^{(State)}$. By Lem. 250 we know $\eta'^{(State)}|_{fv(I)} = (\eta_1{}^{(State)} \oplus_p \eta_2'{}^{(State)})|_{fv(I)} = \eta_1{}^{(State)}|_{fv(I)} \oplus_p \eta_2'{}^{(State)}|_{fv(I)} = \eta_1{}^{(State)}|_{fv(I)} \oplus_p \eta_2{}^{(State)}|_{fv(I)} = (\eta_1{}^{(State)} \oplus_p \eta_2{}^{(State)})|_{fv(I)}$. From $\eta_1{}^{(State)} \oplus_p \eta_2{}^{(State)} \models I$ by

Lem. 272 we know $\eta'^{(State)} \models I$. From $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}), R \vee G_2, I) \Longrightarrow^m_{\text{NST}} (G_1, M)$ for all m by Lem. 241 we have $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}))|_{\mathbf{skip}}{}^{(State)} \models M$. From $0 < p < 1$ by Lem. 275 we know $supp(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)})) = supp(\eta_1) \cup supp(\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}) \supseteq supp(\eta_1)$. By Lem. 277 and Lem 24 we know $supp(\eta_1|_{\mathbf{skip}}{}^{(State)}) \subseteq supp((\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}))|_{\mathbf{skip}}{}^{(State)})$. From $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}))|_{\mathbf{skip}}{}^{(State)} \models M$ and $\mathbf{scl}(M)$ we know $\eta_1|_{\mathbf{skip}}{}^{(State)} \models M$. From $\eta_1{}^{(Stmt)}(\mathbf{skip}) = 1$ by Lem. 199 we know $\eta_1|_{\mathbf{skip}} = \eta_1$, thus $\eta_1{}^{(State)} = \eta_1|_{\mathbf{skip}}{}^{(State)} \models M$. From $R, G_2, \text{true} \models_{\text{NST}} \{M\}C_2\{Q\}$ we know $(\delta(C_2) \otimes \eta_1{}^{(State)}, R, \text{true}) \Longrightarrow^k_{\text{NST}} (G_2, Q)$. From $\mathbf{Nosplit}(C_2)$ by Lem. 280 we know $\mathbf{Nosplit}(\delta(C_2) \otimes \eta_1{}^{(State)})$. By Lem. 282 we know $\mathbf{disablesplit}(\text{true}, \delta(C_2) \otimes \eta_1{}^{(State)})$. From $\mathbf{Nosplit}(\eta_2)$ and $\eta_2 \rightsquigarrow (\theta_2, \eta_2')$ by Lem. 281 we know $\mathbf{Nosplit}(\eta_2')$. By Lem. 282 we know $\mathbf{disablesplit}(\text{true}, \eta_2')$. From $(\delta(C_2) \otimes \eta_1{}^{(State)}, R, \text{true}) \Longrightarrow^k_{\text{NST}} (G_2, Q)$, $(\eta_2', R, \text{true}) \Longrightarrow^k_{\text{NST}} (G_2, Q)$, $\mathbf{closed}(Q)$, $\mathbf{disablesplit}(\text{true}, \delta(C_2) \otimes \eta_1{}^{(State)})$, $\mathbf{disablesplit}(\text{true}, \eta_2')$ and $0 < p < 1$ by Lem. 298 we know $((\delta(C_2) \otimes \eta_1{}^{(State)}) \oplus_p \eta_2', R, \text{true}) \Longrightarrow^k_{\text{NST}} (G_2, Q)$. From $\eta' = (\delta(C_2) \otimes \eta_1{}^{(State)}) \oplus_p \eta_2'$ we know $(\eta', R, I) \Longrightarrow^k_{\text{NST}} (G_2, Q)$. From $\mathbf{Nosplit}(\delta(C_2) \otimes \eta_1{}^{(State)})$ and $\mathbf{Nosplit}(\eta_2')$ by Lem. 283 we know $\mathbf{Nosplit}((\delta(C_2) \otimes \eta_1{}^{(State)}) \oplus_{p_2} \eta_2')$, i.e., $\mathbf{Nosplit}(\eta')$. From $(\eta', R, I) \Longrightarrow^k_{\text{NST}} (G_2, Q)$, $\eta'^{(State)} \models I$ and $\mathbf{Nosplit}(\eta')$ by IH we have $(\eta', R, I) \Longrightarrow^k_{\text{NST}} (G_1 \vee G_2, Q)$.

- $\eta_1{}^{(Stmt)}(\mathbf{skip}) = 0$.
  From $\eta_1{}^{(Stmt)}(\mathbf{skip}) = 0$ and $\eta_1; C_2 \rightsquigarrow (\theta_1, \eta_1')$ by Lem. 213 and we know there exists η_{11}' such that $\eta_1' = \eta_{11}'; C_2$ and $\eta_1 \rightsquigarrow (\theta_1, \eta_{11}')$, thus $\eta' = \eta_1' \oplus_p \eta_2' = \eta_{11}'; C_2 \oplus_p \eta_2'$. By Lem. 192 we know $\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)} \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in supp(\eta_2{}^{(State)})\}, \delta(\mathbf{skip}) \otimes \eta_2{}^{(State)})$. From $\eta_1 \rightsquigarrow (\theta_1, \eta_{11}')$ and $0 < p < 1$ by Lem. 246 we know $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)})) \rightsquigarrow (\theta_1 \cup \{(\sigma, \sigma) \mid \sigma \in supp(\eta_2{}^{(State)})\}, \eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}))$. From $0 < p < 1$ by Lem. 243 and Lem. 190 we know $nextsplit(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)})) = nextsplit(\eta_1) \cup nextsplit(\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}) \supseteq nextsplit(\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}) = nextsplit(\mathbf{skip}) = \{\mathbf{split}(\text{true})\}$. From $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)})) \rightsquigarrow (\theta_1 \cup \{(\sigma, \sigma) \mid \sigma \in supp(\eta_2{}^{(State)})\}, \eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}))$ by Lem. 191 we know $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)})) \hookrightarrow (\theta_1 \cup \{(\sigma, \sigma) \mid \sigma \in supp(\eta_2{}^{(State)})\}, \eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}))$. From $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}), R \vee G_2, I) \Longrightarrow^m_{\text{NST}} (G_1, M)$ for all m by Lem. 241 we know $\theta_1 \cup \{(\sigma, \sigma) \mid \sigma \in supp(\eta_2{}^{(State)})\} \subseteq [\![G_1]\!]$, $(\eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}))^{(State)} \models I$ and $(\eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}), R \vee G_2, I) \Longrightarrow^m_{\text{NST}} (G_1, M)$ for all m, thus $\theta_1 \subseteq \theta_1 \cup \{(\sigma, \sigma) \mid \sigma \in supp(\eta_2{}^{(State)})\} \subseteq [\![G_1]\!]$. By Lem. 12 and Lem. 19 we know $\eta_{11}'{}^{(State)} \oplus_p \eta_2{}^{(State)} = (\eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes \eta_2{}^{(State)}))^{(State)} \models I$. From $\mathbf{Nosplit}(\eta_2)$ by Lem. 244 we know $nextsplit(\eta_2) = \{\mathbf{split}(\text{true})\}$.

From $\eta_2 \rightsquigarrow (\theta_2, \eta_2')$ by Lem. 191 we have $\eta_2 \hookrightarrow (\theta_2, \eta_2')$. From $(\eta_2, R, \text{true}) \Longmapsto^{k+1}_{\text{NST}} (G_2, Q)$ we know $\theta_2 \subseteq [\![G_2]\!]$ and $(\eta_2', R, \text{true}) \Longmapsto^{k}_{\text{NST}} (G_2, Q)$, thus $\theta = \theta_1 \cup \theta_2 \subseteq [\![G_1]\!] \cup [\![G_2]\!] = [\![G_1 \vee G_2]\!]$. From $\theta_2 \subseteq [\![G_2]\!]$ and $\forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x)$ we know $\forall x \in fv(I), (\sigma, \sigma') \in \theta.\ \sigma'(x) = \sigma(x)$. From $\eta_2 \rightsquigarrow (\theta_2, \eta_2')$ by Lem. 249 we know ${\eta_2'}^{(State)}|_{fv(I)} = {\eta_2}^{(State)}|_{fv(I)}$. From $\eta' = \eta_{11}'; C_2 \oplus_p \eta_2'$ by Lem. 12 and Lem. 201 we know ${\eta'}^{(State)} = {\eta_{11}'}; {C_2}^{(State)} \oplus_p {\eta_2'}^{(State)} = {\eta_{11}'}^{(State)} \oplus_p {\eta_2'}^{(State)}$. By Lem. 250 we know ${\eta'}^{(State)}|_{fv(I)} = ({\eta_{11}'}^{(State)} \oplus_p {\eta_2'}^{(State)})|_{fv(I)} = {\eta_{11}'}^{(State)}|_{fv(I)} \oplus_p {\eta_2'}^{(State)}|_{fv(I)} = {\eta_{11}'}^{(State)}|_{fv(I)} \oplus_p {\eta_2}^{(State)}|_{fv(I)} = ({\eta_{11}'}^{(State)} \oplus_p {\eta_2}^{(State)})|_{fv(I)}$. From ${\eta_{11}'}^{(State)} \oplus_p {\eta_2}^{(State)} \models I$ by Lem. 272 we know ${\eta'}^{(State)} \models I$. From $\mathbf{Id} \Rightarrow R \vee G_2$ by Lem. 273 we know $\eta_{11}' \overset{R \vee G_2}{\mapsto} \eta_{11}'$. From $\eta_2 \rightsquigarrow (\theta_2, \eta_2')$ and $\theta_2 \subseteq [\![G_2]\!] \subseteq [\![R \vee G_2]\!]$ by Lem. 274 we know $\eta_2 \overset{R \vee G_2}{\mapsto} \eta_2'$. By Lem. 178 we know ${\eta_2}^{(State)} \overset{R \vee G_2}{\mapsto} {\eta_2'}^{(State)}$. By Lem. 240 we know $\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)} \overset{R \vee G_2}{\mapsto} \delta(\mathbf{skip}) \otimes {\eta_2'}^{(State)}$.

From $\eta_{11}' \overset{R}{\mapsto} \eta_{11}'$ and $0 < p < 1$ by Lem. 246 we know $(\eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})) \overset{R}{\mapsto} \eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2'}^{(State)})$. By Lem. 12 we know $(\eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2'}^{(State)}))^{(State)} = {\eta_{11}'}^{(State)} \oplus_p {\eta_2'}^{(State)} = {\eta'}^{(State)} \models I$. By Lem. 171 we know $(\eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2'}^{(State)}))|_{\text{true}} = \eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2'}^{(State)})$. From $(\eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})) \overset{R}{\mapsto} \eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2'}^{(State)})$ and $(\eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2'}^{(State)}))^{(State)} \models I$ we have $(\eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})) \underset{I}{\overset{R}{\rightarrowtail}} \eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2'}^{(State)})$. From $(\eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}), R \vee G_2, I) \Longmapsto^{m}_{\text{NST}} (G_1, M)$ for all m by Lem. 241 we have $(\eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2'}^{(State)}), R \vee G_2, I) \Longmapsto^{m}_{\text{NST}} (G_1, M)$ for all m. From $\mathbf{Nosplit}(\eta_2)$ and $\eta_2 \rightsquigarrow (\theta_2, \eta_2')$ by Lem. 281 we know $\mathbf{Nosplit}(\eta_2')$. From $\eta' = \eta_{11}''; C_2 \oplus_p \eta_2'$, $0 < p_1 < 1$, $(\eta_2', R, \text{true}) \Longmapsto^{k}_{\text{NST}} (G_2, Q)$, $\mathbf{Nosplit}(\eta_2')$ and $(\eta_{11}' \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2'}^{(State)}), R \vee G_2, I) \Longmapsto^{m}_{\text{NST}} (G_1, M)$ for all m by IH we have $(\eta', R, I) \Longmapsto^{k}_{\text{NST}} (G_1 \vee G_2, Q)$.

- $0 < {\eta_1}^{(Stmt)}(\mathbf{skip}) < 1$.

  Let $p' \overset{\text{def}}{=} {\eta_1}^{(Stmt)}(\mathbf{skip})$, then $0 < p' < 1$. By Lem. 247 there exists η_{11}, η_{12} such that $\eta_1 = \eta_{11} \oplus_{p'} \eta_{12}$, ${\eta_{11}}^{(Stmt)}(\mathbf{skip}) = 1$ and ${\eta_{12}}^{(Stmt)}(\mathbf{skip}) = 0$, by Lem. 248 we know $\eta_1; C_2 = (\eta_{11} \oplus_{p'} \eta_{12}); C_2 = \eta_{11}; C_2 \oplus_{p'} \eta_{12}; C_2$. From $\eta_1; C_2 \rightsquigarrow (\theta_1, \eta_1')$ we know $\eta_{11}; C_2 \oplus_{p'} \eta_{12}; C_2 \rightsquigarrow (\theta_1, \eta_1')$. From $0 < p' < 1$ by Lem. 245 there exists $\theta_{11}, \theta_{12}, \eta_{11}', \eta_{12}'$ such that $\eta_1' = \eta_{11}' \oplus_{p'} \eta_{12}'$, $\theta_1 = \theta_{11} \cup \theta_{12}$, $\eta_{11}; C_2 \rightsquigarrow (\theta_{11}, \eta_{11}')$ and $\eta_{12}; C_2 \rightsquigarrow (\theta_{12}, \eta_{12}')$. From

${\eta_{11}}^{(Stmt)}(\mathbf{skip}) = 1$ by Lem. 13 we know $\eta_{11} = \delta(\mathbf{skip}) \otimes {\eta_{11}}^{(State)}$. By Lem. 18 we know ${\eta_{11}}^{(Stmt)} = \delta(\mathbf{skip})$. From $\eta_{11}; C_2 \rightsquigarrow (\theta_{11}, \eta'_{11})$ by Lem. 210 and Lem. 193 we know $\theta_{11} = \{(\sigma, \sigma) \mid \sigma \in supp({\eta_{11}}^{(State)})\} \subseteq [\![\mathbf{Id}]\!]$ and $\eta'_{11} = \delta(C_2) \otimes {\eta_{11}}^{(State)}$. By Lem. 192 we know $\eta_{11} \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in supp({\eta_{11}}^{(State)})\}, \delta(\mathbf{skip}) \otimes {\eta_{11}}^{(State)})$, i.e., $\eta_{11} \rightsquigarrow (\theta_{11}, \eta_{11})$. From ${\eta_{12}}^{(Stmt)}(\mathbf{skip}) = 0$ and $\eta_{12}; C_2 \rightsquigarrow (\theta_{12}, \eta'_{12})$ by Lem. 213 and we know there exists η''_{12} such that $\eta'_{12} = \eta''_{12}; C_2$ and $\eta_{12} \rightsquigarrow (\theta_{12}, \eta''_{12})$. From $\eta_{11} \rightsquigarrow (\theta_{11}, \eta_{11})$ and $0 < p' < 1$ by Lem. 246 we know $\eta_{11} \oplus_{p'} \eta_{12} \rightsquigarrow (\theta_{11} \cup \theta_{12}, \eta_{11} \oplus_{p'} \eta''_{12})$, i.e., $\eta_1 \rightsquigarrow (\theta_1, \eta_{11} \oplus_{p'} \eta''_{12})$. By Lem. 192 we know $\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)} \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in supp({\eta_2}^{(State)})\}, \delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})$. From $\eta_1 \rightsquigarrow (\theta_1, \eta_{11} \oplus_{p'} \eta''_{12})$ and $0 < p < 1$ by Lem. 246 we know $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})) \rightsquigarrow (\theta_1 \cup \{(\sigma, \sigma) \mid \sigma \in supp({\eta_2}^{(State)})\}, (\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}))$. From $0 < p < 1$ by Lem. 243 and Lem. 190 we know $nextsplit(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})) = nextsplit(\eta_1) \cup nextsplit(\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}) \supseteq nextsplit(\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}) = nextsplit(\mathbf{skip}) = \{\mathbf{split}(\mathrm{true})\}$. From $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})) \rightsquigarrow (\theta_1 \cup \{(\sigma, \sigma) \mid \sigma \in supp({\eta_2}^{(State)})\}, (\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}))$ by Lem. 191 we know $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})) \hookrightarrow (\theta_1 \cup \{(\sigma, \sigma) \mid \sigma \in supp({\eta_2}^{(State)})\}, (\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}))$. From $(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}), R \vee G_2, I) \Longrightarrow^m_{\mathrm{NST}} (G_1, M)$ for all m by Lem. 241 we know $\theta_1 \cup \{(\sigma, \sigma) \mid \sigma \in supp({\eta_2}^{(State)})\} \subseteq [\![G_1]\!]$,
$((\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}))^{(State)} \models I$ and $((\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}),$
$R \vee G_2, I) \Longrightarrow^m_{\mathrm{NST}} (G_1, M)$ for all m, thus $\theta_1 \subseteq \theta_1 \cup \{(\sigma, \sigma) \mid \sigma \in supp({\eta_2}^{(State)})\} \subseteq [\![G_1]\!]$. By Lem. 12 and Lem. 19 we know $({\eta_{11}}^{(State)} \oplus_{p'} {\eta''_{12}}^{(State)}) \oplus_p {\eta_2}^{(State)} =$
$((\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}))^{(State)} \models I$. From $\mathbf{Nosplit}(\eta_2)$ by Lem. 244 we know $nextsplit(\eta_2) = \{\mathbf{split}(\mathrm{true})\}$. From $\eta_2 \rightsquigarrow (\theta_2, \eta'_2)$ by Lem. 191 we have $\eta_2 \hookrightarrow (\theta_2, \eta'_2)$. From $(\eta_2, R, \mathrm{true}) \Longrightarrow^{k+1}_{\mathrm{NST}} (G_2, Q)$ we know $\theta_2 \subseteq [\![G_2]\!]$ and $(\eta'_2, R, \mathrm{true}) \Longrightarrow^k_{\mathrm{NST}} (G_2, Q)$, thus $\theta = \theta_1 \cup \theta_2 \subseteq [\![G_1]\!] \cup [\![G_2]\!] = [\![G_1 \vee G_2]\!]$. From $\theta_2 \subseteq [\![G_2]\!]$ and $\forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x)$ we know $\forall x \in fv(I), (\sigma, \sigma') \in \theta.\ \sigma'(x) = \sigma(x)$. From $\eta_2 \rightsquigarrow (\theta_2, \eta'_2)$ by Lem. 249 we know ${\eta'_2}^{(State)}|_{fv(I)} = {\eta_2}^{(State)}|_{fv(I)}$. From $\eta' = \eta'_1 \oplus_p \eta'_2 = ((\eta'_{11} \oplus_{p'} \eta'_{12}) \oplus_p \eta'_2) = (((\delta(C_2) \otimes {\eta_{11}}^{(State)}) \oplus_{p'} \eta''_{12}; C_2) \oplus_p \eta'_2)$ by Lem. 12 we know ${\eta'}^{(State)} = ({\eta_{11}}^{(State)} \oplus_{p'} {\eta''_{12}}^{(State)}) \oplus_p {\eta'_2}^{(State)}$. By Lem. 250 we know ${\eta'}^{(State)}|_{fv(I)} = (({\eta_{11}}^{(State)} \oplus_{p'} {\eta''_{12}}^{(State)}) \oplus_p {\eta'_2}^{(State)})|_{fv(I)} = (({\eta_{11}}^{(State)} \oplus_{p'} {\eta''_{12}}^{(State)})|_{fv(I)} \oplus_p {\eta'_2}^{(State)}|_{fv(I)} = (({\eta_{11}}^{(State)} \oplus_{p'} {\eta''_{12}}^{(State)})|_{fv(I)} \oplus_p {\eta_2}^{(State)}|_{fv(I)} = (({\eta_{11}}^{(State)} \oplus_{p'} {\eta''_{12}}^{(State)}) \oplus_p {\eta_2}^{(State)})|_{fv(I)}$. From $({\eta_{11}}^{(State)} \oplus_{p'} {\eta''_{12}}^{(State)}) \oplus_p {\eta_2}^{(State)} \models I$ by Lem. 272 we know ${\eta'}^{(State)} \models I$. From $\mathbf{Id} \Rightarrow R \vee G_2$ by Lem. 273 we know $\eta_{11} \oplus_{p'}$

$\eta''_{12} \overset{R\vee G_2}{\mapsto} \eta_{11} \oplus_{p'} \eta''_{12}$. From $\eta_2 \rightsquigarrow (\theta_2, \eta'_2)$ and $\theta_2 \subseteq [\![G_2]\!] \subseteq [\![R \vee G_2]\!]$ by Lem. 274 we know $\eta_2 \overset{R\vee G_2}{\mapsto} \eta'_2$. By Lem. 178 we know ${\eta_2}^{(State)} \overset{R\vee G_2}{\mapsto} {\eta'_2}^{(State)}$. By Lem. 240 we know $\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)} \overset{R\vee G_2}{\mapsto} \delta(\mathbf{skip}) \otimes {\eta'_2}^{(State)}$. From $\eta_{11} \oplus_{p'} \eta''_{12} \overset{R\vee G_2}{\mapsto} \eta_{11} \oplus_{p'} \eta''_{12}$ and $0 < p < 1$ by Lem. 246 we know $((\eta_{11} \oplus_{p'} \eta_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})) \overset{R\vee G_2}{\mapsto} (\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta'_2}^{(State)})$. By Lem. 12 we know $((\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta'_2}^{(State)}))^{(State)} = ({\eta_{11}}^{(State)} \oplus_{p'} {\eta''_{12}}^{(State)}) \oplus_p {\eta'_2}^{(State)} = {\eta'}^{(State)} \models I$. By Lem. 171 we know $((\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta'_2}^{(State)}))|_{\mathrm{true}} = (\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta'_2}^{(State)})$. From $((\eta_{11} \oplus_{p'} \eta_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})) \overset{R\vee G_2}{\mapsto} (\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta'_2}^{(State)})$ and $((\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta'_2}^{(State)}))^{(State)} \models I$ we have $((\eta_{11} \oplus_{p'} \eta_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})) \underset{I}{\overset{R\vee G_2}{\rightarrowtail}} (\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta'_2}^{(State)})$. From $((\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}), R \vee G_2, I) \Longrightarrow^m_{\mathrm{NST}} (G_1, M)$ for all m by Lem. 241 we have $((\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}))|_{\mathbf{skip}}{}^{(State)} \models M$ and $((\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta'_2}^{(State)}), R \vee G_2, I) \Longrightarrow^m_{\mathrm{NST}} (G_1, M)$ for all m. From $0 < p < 1$ and $0 < p' < 1$ by Lem. 275 we know $supp((\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})) = supp(\eta_{11}) \cup supp(\eta''_{12}) \cup supp(\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})) \supseteq supp(\eta_{11})$. By Lem. 277 and Lem 24 we know $supp({\eta_{11}|_{\mathbf{skip}}}^{(State)}) \subseteq supp(((\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}))|_{\mathbf{skip}}{}^{(State)})$. From $((\eta_{11} \oplus_{p'} \eta''_{12}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}))|_{\mathbf{skip}}{}^{(State)} \models M$ and $\mathbf{scl}(M)$ we know ${\eta_{11}|_{\mathbf{skip}}}^{(State)} \models M$. From ${\eta_{11}}^{(Stmt)}(\mathbf{skip}) = 1$ by Lem. 199 we know $\eta_{11}|_{\mathbf{skip}} = \eta_{11}$, thus ${\eta_{11}}^{(State)} = {\eta_{11}|_{\mathbf{skip}}}^{(State)} \models M$. From $R, G_2, \mathrm{true} \models_{\mathrm{NST}} \{M\} C_2 \{Q\}$ we know $(\delta(C_2) \otimes {\eta_{11}}^{(State)}, R, \mathrm{true}) \Longrightarrow^k_{\mathrm{NST}} (G_2, Q)$. Let $p_1 \overset{\mathrm{def}}{=} p \cdot (1 - p')$ and $p_2 \overset{\mathrm{def}}{=} \frac{p \cdot p'}{1 - p \cdot (1 - p')}$. From $0 < p < 1$ and $0 < p' < 1$ we know $0 < p_1 < 1$ and $0 < p_2 < 1$. By Lem. 278 and Lem. 279 we know $\eta' = (((\delta(C_2) \otimes {\eta_{11}}^{(State)}) \oplus_{p'} \eta''_{12}; C_2) \oplus_p \eta'_2) = ((\eta''_{12}; C_2 \oplus_{1-p'} (\delta(C_2) \otimes {\eta_{11}}^{(State)})) \oplus_p \eta'_2) = \eta''_{12}; C_2 \oplus_{p_1} ((\delta(C_2) \otimes {\eta_{11}}^{(State)}) \oplus_{p_2} \eta'_2)$. From $\mathbf{Nosplit}(C_2)$ by Lem. 280 we know $\mathbf{Nosplit}(\delta(C_2) \otimes {\eta_{11}}^{(State)})$. By Lem. 282 we know $\mathbf{disablesplit}(\mathrm{true}, \delta(C_2) \otimes {\eta_{11}}^{(State)})$. From $\mathbf{Nosplit}(\eta_2)$ and $\eta_2 \rightsquigarrow (\theta_2, \eta'_2)$ by Lem. 281 we know $\mathbf{Nosplit}(\eta'_2)$. By Lem. 282 we know $\mathbf{disablesplit}(\mathrm{true}, \eta'_2)$. From $(\delta(C_2) \otimes {\eta_{11}}^{(State)}, R, \mathrm{true}) \Longrightarrow^k_{\mathrm{NST}} (G_2, Q)$, $(\eta'_2, R, \mathrm{true}) \Longrightarrow^k_{\mathrm{NST}} (G_2, Q)$, $\mathbf{closed}(Q)$, $\mathbf{disablesplit}(\mathrm{true}, \delta(C_2) \otimes {\eta_{11}}^{(State)})$, $\mathbf{disablesplit}(\mathrm{true}, \eta'_2)$ and $0 < p_2 < 1$ by Lem. 298 we know $((\delta(C_2) \otimes {\eta_{11}}^{(State)}) \oplus_{p_2} \eta'_2, R, \mathrm{true}) \Longrightarrow^k_{\mathrm{NST}} (G_2, Q)$. From $\mathbf{Nosplit}(\delta(C_2) \otimes$

${\eta_{11}}^{(State)}$) and $\mathbf{Nosplit}(\eta_2')$ by Lem. 283 we know $\mathbf{Nosplit}((\delta(C_2)\otimes {\eta_{11}}^{(State)}) \oplus_{p_2} \eta_2')$. From $\eta_{11} = \delta(\mathbf{skip}) \otimes {\eta_{11}}^{(State)}$ we know

$$
\begin{aligned}
&(\eta_{11} \oplus_{p'} \eta_{12}'') \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2'}^{(State)}) \\
&= ((\delta(\mathbf{skip}) \otimes \eta_{11}) \oplus_{p'} \eta_{12}'') \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2'}^{(State)}) \\
&= (\eta_{12}'' \oplus_{1-p'} (\delta(\mathbf{skip}) \otimes \eta_{11})) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2'}^{(State)}) \qquad \text{(by Lem. 278)} \\
&= \eta_{12}'' \oplus_{p_1} ((\delta(\mathbf{skip}) \otimes {\eta_{11}}^{(State)}) \oplus_{p_2} (\delta(\mathbf{skip}) \otimes {\eta_2'}^{(State)})) \qquad \text{(by Lem. 279)} \\
&= \eta_{12}'' \oplus_{p_1} (\delta(\mathbf{skip}) \otimes ({\eta_{11}}^{(State)} \oplus_{p_2} {\eta_2'}^{(State)})) \qquad \text{(by Lem.14)} \\
&= \eta_{12}'' \oplus_{p_1} (\delta(\mathbf{skip}) \otimes (({\delta(C_2) \otimes {\eta_{11}}^{(State)})}^{(State)} \oplus_{p_2} {\eta_2'}^{(State)}) \qquad \text{(by Lem. 19)} \\
&= \eta_{12}'' \oplus_{p_1} (\delta(\mathbf{skip}) \otimes ((\delta(C_2) \otimes {\eta_{11}}^{(State)}) \oplus_{p_2} \eta_2')^{(State)}). \qquad \text{(by Lem. 12)}
\end{aligned}
$$

From $((\eta_{11} \oplus_{p'} \eta_{12}'') \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2'}^{(State)}), R \vee G_2, I) \Longmapsto_{\text{NST}}^{m} (G_1, M)$ for all m we know $(\eta_{12}'' \oplus_{p_1} (\delta(\mathbf{skip}) \otimes ((\delta(C_2) \otimes {\eta_{11}}^{(State)}) \oplus_{p_2} \eta_2')^{(State)}), R \vee G_2, I) \Longmapsto_{\text{NST}}^{m} (G_1, M)$ for all m. From $\eta' = \eta_{12}''; C_2 \oplus_{p_1} ((\delta(C_2) \otimes {\eta_{11}}^{(State)}) \oplus_{p_2} \eta_2'), 0 < p_1 < 1, ((\delta(C_2) \otimes {\eta_{11}}^{(State)}) \oplus_{p_2} \eta_2', R, \text{true}) \Longmapsto_{\text{NST}}^{k} (G_2, Q)$ and $\mathbf{Nosplit}((\delta(C_2) \otimes {\eta_{11}}^{(State)}) \oplus_{p_2} \eta_2')$ by IH we have $(\eta', R, I) \Longmapsto_{\text{NST}}^{k} (G_1 \vee G_2, Q)$.

- $(\eta, R, \text{true}) \Longmapsto_{\text{NST}}^{k+1} (G_2, Q)$, $\eta^{(State)} \models I$ and $\mathbf{Nosplit}(\eta)$.
  To prove $(\eta, R, I) \Longmapsto_{\text{NST}}^{k+1} (G_1 \vee G_2, Q)$, we need to prove
  - if $\eta^{(Stmt)}(\mathbf{skip}) > 0$, then ${\eta|_{\mathbf{skip}}}^{(State)} \models Q$.
    From $(\eta, R, \text{true}) \Longmapsto_{\text{NST}}^{k+1} (G_2, Q)$ and $\eta^{(Stmt)}(\mathbf{skip}) > 0$ we know ${\eta|_{\mathbf{skip}}}^{(State)} \models Q$.
  - $\eta^{(State)} \models I$.
    By assumption.
  - for all η', if $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longmapsto_{\text{NST}}^{k} (G_1 \vee G_2, Q)$.
    For all η' such that $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, there exists η'' and b such that $\eta \overset{R}{\mapsto} \eta''$, $\eta''|_b = \eta'$ and ${\eta'}^{(State)} \models I$, thus $\eta \overset{R}{\underset{\text{true}}{\rightarrowtail}} \eta'$. From $(\eta, R, \text{true}) \Longmapsto_{\text{NST}}^{k+1} (G_2, Q)$ we know $(\eta', R, \text{true}) \Longmapsto_{\text{NST}}^{k} (G_2, Q)$. From $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$ by Lem. 188 we know ${\eta'}^{(Stmt)} \subseteq \eta^{(Stmt)}$. From $\mathbf{Nosplit}(\eta)$ by Lem. 239 we know $\mathbf{Nosplit}(\eta')$. From $(\eta', R, \text{true}) \Longmapsto_{\text{NST}}^{k} (G_2, Q)$, ${\eta'}^{(State)} \models I$ and $\mathbf{Nosplit}(\eta')$ by IH we have $(\eta', R, I) \Longmapsto_{\text{NST}}^{k} (G_1 \vee G_2, Q)$.
  - for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G_1 \vee G_2]\!]$, ${\eta'}^{(State)} \models I$ and $(\eta', R, I) \Longmapsto_{\text{NST}}^{k} (G_1 \vee G_2, Q)$.
    For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, from $(\eta, R, \text{true}) \Longmapsto_{\text{NST}}^{k+1} (G_2, Q)$ we know $\theta \subseteq [\![G_2]\!] \subseteq [\![G_1 \vee G_2]\!]$, $(\eta', R, \text{true}) \Longmapsto_{\text{NST}}^{k} (G_2, Q)$. From $\mathbf{Nosplit}(\eta)$ by Lem. 244 we know $\mathit{nextsplit}(\eta) = \{\mathbf{split}(\text{true})\}$. From $\eta \hookrightarrow (\theta, \eta')$ by Lem. 191 we have $\eta \rightsquigarrow (\theta, \eta')$. From $\theta \subseteq [\![G_2]\!]$ and $\forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x)$ we know $\forall x \in fv(I), (\sigma, \sigma') \in \theta.\ \sigma'(x) = \sigma(x)$. By Lem. 249 we know ${\eta'}^{(State)}|_{fv(I)} = {\eta^{(State)}}|_{fv(I)}$. From $\eta^{(State)} \models I$ by Lem. 272 we know ${\eta'}^{(State)} \models I$. From $\mathbf{Nosplit}(\eta)$

and $\eta \rightsquigarrow (\theta, \eta')$ by Lem. 281 we know $\mathbf{Nosplit}(\eta')$. From $(\eta', R, \text{true}) \Longmapsto^{k}_{\text{NST}} (G_2, Q), {\eta'}^{(State)} \models I$ and $\mathbf{Nosplit}(\eta')$ by IH we have $(\eta', R, I) \Longmapsto^{k}_{\text{NST}} (G_1 \vee G_2, Q)$.

Lemma 285. *For all η, if $0 < [\![\boldsymbol{Pr}(b)]\!]_{\eta^{(State)}} < 1$, then there exists η_1 and η_2 such that $\eta = \eta_1 \oplus_{[\![\boldsymbol{Pr}(b)]\!]_{\eta^{(State)}}} \eta_2$, ${\eta_1}^{(State)} \models \lceil b \rceil$ and ${\eta_2}^{(State)} \models \lceil \neg b \rceil$.*

Proof. For all η such that $0 < [\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}} < 1$, let $\eta_1 \stackrel{\text{def}}{=} \lambda(C,\sigma).\frac{\chi(\sigma \models b)\cdot\eta(C,\sigma)}{[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}}$ and $\eta_2 \stackrel{\text{def}}{=} \lambda(C,\sigma).\frac{\chi(\sigma \not\models b)\cdot\eta(C,\sigma)}{1-[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}}$, then

$$
\begin{aligned}
&\eta_1 \oplus_{[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}} \eta_2 \\
&= \lambda(C,\sigma).\ [\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}} \cdot \eta_1(C,\sigma) + (1 - [\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}) \cdot \eta_2(C,\sigma) \\
&= \lambda(C,\sigma).\ [\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}} \cdot \tfrac{\chi(\sigma \models b)\cdot\eta(C,\sigma)}{[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}} + (1 - [\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}) \cdot \tfrac{\chi(\sigma \not\models b)\cdot\eta(C,\sigma)}{1-[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}} \\
&= \lambda(C,\sigma).\ \chi(\sigma \models b) \cdot \eta(C,\sigma) + \chi(\sigma \not\models b) \cdot \eta(C,\sigma) \\
&= \lambda(C,\sigma).\ \eta(C,\sigma) \\
&= \eta.
\end{aligned}
$$

For all $\sigma \in \mathit{supp}({\eta_1}^{(State)})$, by Lem. 22 we know $\mathit{supp}({\eta_1}^{(State)}) = \mathit{range}(\mathit{supp}(\eta_1))$, thus $\sigma \in \mathit{range}(\mathit{supp}(\eta_1))$, so there exists C such that $\eta_1(C,\sigma) > 0$, i.e., $\frac{\chi(\sigma \models b)\cdot\eta(C,\sigma)}{[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}} > 0$, so $\sigma \models b$. Therefore ${\eta_1}^{(State)} \models \lceil b \rceil$.
For all $\sigma \in \mathit{supp}({\eta_2}^{(State)})$, by Lem. 22 we know $\mathit{supp}({\eta_2}^{(State)}) = \mathit{range}(\mathit{supp}(\eta_2))$, thus $\sigma \in \mathit{range}(\mathit{supp}(\eta_2))$, so there exists C such that $\eta_2(C,\sigma) > 0$, i.e., $\frac{\chi(\sigma \not\models b)\cdot\eta(C,\sigma)}{1-[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}} > 0$, so $\sigma \not\models b$, i.e., $\sigma \models \neg b$. Therefore ${\eta_2}^{(State)} \models \lceil b \rceil$.

Lemma 286. *For all Q, C, μ, if $\mathbf{disablesplit}(Q, C)$, then $\mathbf{disablesplit}(Q, \delta(C) \otimes \mu)$.*

Proof. For all Q, C, μ such that $\mathbf{disablesplit}(Q, C)$, To prove $\mathbf{disablesplit}(Q, \delta(C) \otimes \mu)$, we need to prove $\mathbf{disablesplit}(Q, C')$ for all $C' \in \mathit{supp}((\delta(C) \otimes \mu)^{(State)})$. For all $C' \in \mathit{supp}((\delta(C) \otimes \mu)^{(Stmt)})$, by Lem. 18 we know $(\delta(C) \otimes \mu)^{(Stmt)} = \delta(C)$, thus $C' \in \mathit{supp}(\delta(C)) = \{C\}$, so $C' = C$. From $\mathbf{disablesplit}(Q, C)$ we have $\mathbf{disablesplit}(Q, C')$.

Lemma 287. *For all η and b, $\eta|_{\boldsymbol{skip}}$ exists if and only if $\eta^{(Stmt)}(\boldsymbol{skip}) > 0$.*

Proof. For all η, by definition of $\eta|_{\mathbf{skip}}$ we know $\eta|_{\mathbf{skip}}$ exists if and only if $\eta|_{\lambda(C,\sigma).\ C=\mathbf{skip}}$ exists. By Eqn. 2 we know $\eta|_{\lambda(C,\sigma).\ C=\mathbf{skip}}$ exists if and only if $\mathbf{Pr}_{(C,\sigma)\sim W}[C = \mathbf{skip}] > 0$, i.e., $\eta^{(Stmt)}(\mathbf{skip}) > 0$. Therefore, $\eta|_{\mathbf{skip}}$ exists if and only if $\eta^{(Stmt)}(\mathbf{skip}) > 0$.

Lemma 288. *For all η_1, η_2, p, if $0 < p < 1$, ${\eta_1}^{(Stmt)}(\boldsymbol{skip}) = p_1$, and ${\eta_2}^{(Stmt)}(\boldsymbol{skip}) = p_2$, then $(\eta_1 \oplus_p \eta_2)|_{\boldsymbol{skip}} =$*

$$
\begin{cases}
\eta_1|_{\boldsymbol{skip}} \oplus_{\frac{p\cdot p_1}{p\cdot p_1 + (1-p)\cdot p_2}} \eta_2|_{\boldsymbol{skip}}, & \text{if } p_1 > 0 \wedge p_2 > 0 \\
\eta_1|_{\boldsymbol{skip}}, & \text{if } p_1 > 0 \wedge p_2 = 0 \\
\eta_2|_{\boldsymbol{skip}}, & \text{if } p_1 = 0 \wedge p_2 > 0 \\
\textit{undefined}, & \textit{otherwise.}
\end{cases}
$$

Proof. For all η_1, η_2, p, b such that $0 < p < 1$, ${\eta_1}^{(Stmt)}(\mathbf{skip}) = p_1$ and ${\eta_2}^{(Stmt)}(\mathbf{skip}) = p_2$, we prove the four cases respectively.

- $p_1 > 0 \wedge p_2 > 0$.
  By Lem. 12 we know $(\eta_1 \oplus_p \eta_2)^{(Stmt)}(\mathbf{skip}) = ({\eta_1}^{(Stmt)} \oplus_p {\eta_2}^{(Stmt)})(\mathbf{skip}) = p \cdot {\eta_1}^{(Stmt)}(\mathbf{skip}) + (1-p) \cdot {\eta_2}^{(Stmt)}(\mathbf{skip}) = p \cdot p_1 + (1-p) \cdot p_2 > 0$, thus

$$
\begin{aligned}
&(\eta_1 \oplus_p \eta_2)|_{\mathbf{skip}} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(C=\mathbf{skip})\cdot(\eta_1\oplus_p\eta_2)(\mathbf{skip},\sigma)}{(\eta_1\oplus_p\eta_2)^{(Stmt)}(\mathbf{skip})} \qquad \text{(by Lem. 165)} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(C=\mathbf{skip})\cdot(p\cdot\eta_1(\mathbf{skip},\sigma)+(1-p)\cdot\eta_2(\mathbf{skip},\sigma))}{p\cdot p_1+(1-p)\cdot p_2} \\
&= \lambda(C,\sigma).\ \tfrac{p\cdot p_1}{p\cdot p_1+(1-p)\cdot p_2}\cdot\tfrac{\chi(C=\mathbf{skip})\cdot\eta_1(\mathbf{skip},\sigma)}{p_1} + \tfrac{(1-p)\cdot p_2}{p\cdot p_1+(1-p)\cdot p_2}\cdot\tfrac{\chi(C=\mathbf{skip})\cdot\eta_2(\mathbf{skip},\sigma)}{p_2} \\
&= \lambda(C,\sigma).\ \tfrac{p\cdot p_1}{p\cdot p_1+(1-p)\cdot p_2}\cdot\tfrac{\chi(C=\mathbf{skip})\cdot\eta_1(\mathbf{skip},\sigma)}{{\eta_1}^{(Stmt)}(\mathbf{skip})} + (1-\tfrac{p\cdot p_1}{p\cdot p_1+(1-p)\cdot p_2})\cdot\tfrac{\chi(C=\mathbf{skip})\cdot\eta_2(\mathbf{skip},\sigma)}{{\eta_1}^{(Stmt)}(\mathbf{skip})} \\
&= \lambda(C,\sigma).\ \tfrac{p\cdot p_1}{p\cdot p_1+(1-p)\cdot p_2}\cdot\eta_1|_{\mathbf{skip}}(C,\sigma) + (1-\tfrac{p\cdot p_1}{p\cdot p_1+(1-p)\cdot p_2})\cdot\eta_1|_{\mathbf{skip}}(C,\sigma) \qquad \text{(by Lem. 165)} \\
&= \eta_1|_{\mathbf{skip}} \oplus_{\frac{p\cdot p_1}{p\cdot p_1+(1-p)\cdot p_2}} \eta_2|_{\mathbf{skip}}.
\end{aligned}
$$

- $p_1 > 0 \wedge p_2 = 0$.
  By Lem. 12 we know $(\eta_1 \oplus_p \eta_2)^{(Stmt)}(\mathbf{skip}) = ({\eta_1}^{(Stmt)} \oplus_p {\eta_2}^{(Stmt)})(\mathbf{skip}) = p \cdot {\eta_1}^{(Stmt)}(\mathbf{skip}) + (1-p) \cdot {\eta_2}^{(Stmt)}(\mathbf{skip}) = p \cdot p_1 + (1-p) \cdot p_2 = p \cdot p_1 > 0$.
  From $0 = p_2 = {\eta_2}^{(Stmt)}(\mathbf{skip}) = \sum_\sigma \eta_2(\mathbf{skip}, \sigma)$ we know $\eta_2(\mathbf{skip}, \sigma) = 0$ for all σ, thus

$$
\begin{aligned}
&(\eta_1 \oplus_p \eta_2)|_{\mathbf{skip}} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(C=\mathbf{skip})\cdot(\eta_1\oplus_p\eta_2)(\mathbf{skip},\sigma)}{[\![\mathbf{Pr}(b)]\!]_{(\eta_1\oplus_p\eta_2)^{(State)}}} \qquad \text{(by Lem. 165)} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(C=\mathbf{skip})\cdot(p\cdot\eta_1(\mathbf{skip},\sigma)+(1-p)\cdot\eta_2(\mathbf{skip},\sigma))}{p\cdot p_1} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(C=\mathbf{skip})\cdot p\cdot\eta_1(\mathbf{skip},\sigma)}{p\cdot p_1} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(C=\mathbf{skip})\cdot\eta_1(\mathbf{skip},\sigma)}{{\eta_1}^{(Stmt)}(\mathbf{skip})} \\
&= \eta_1|_{\mathbf{skip}}. \qquad \text{(by Lem. 165)}
\end{aligned}
$$

- $p_1 = 0 \wedge p_2 > 0$.
  By Lem. 12 we know $(\eta_1 \oplus_p \eta_2)^{(Stmt)}(\mathbf{skip}) = ({\eta_1}^{(Stmt)} \oplus_p {\eta_2}^{(Stmt)})(\mathbf{skip}) = p \cdot {\eta_1}^{(Stmt)}(\mathbf{skip}) + (1-p) \cdot {\eta_2}^{(Stmt)}(\mathbf{skip}) = p \cdot p_1 + (1-p) \cdot p_2 = (1-p) \cdot p_2 > 0$.
  From $0 = p_1 = {\eta_1}^{(Stmt)}(\mathbf{skip}) = \sum_\sigma \eta_1(\mathbf{skip}, \sigma)$ we know $\eta_1(\mathbf{skip}, \sigma) = 0$ for all σ, thus

$$
\begin{aligned}
&(\eta_1 \oplus_p \eta_2)|_{\mathbf{skip}} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(C=\mathbf{skip})\cdot(\eta_1\oplus_p\eta_2)(\mathbf{skip},\sigma)}{[\![\mathbf{Pr}(b)]\!]_{(\eta_1\oplus_p\eta_2)^{(State)}}} \qquad \text{(by Lem. 165)} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(C=\mathbf{skip})\cdot(p\cdot\eta_1(\mathbf{skip},\sigma)+(1-p)\cdot\eta_2(\mathbf{skip},\sigma))}{p\cdot p_1} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(C=\mathbf{skip})\cdot(1-p)\cdot\eta_2(\mathbf{skip},\sigma)}{(1-p)\cdot p_2} \\
&= \lambda(C,\sigma).\ \tfrac{\chi(C=\mathbf{skip})\cdot\eta_2(\mathbf{skip},\sigma)}{{\eta_2}^{(Stmt)}(\mathbf{skip})} \\
&= \eta_2|_{\mathbf{skip}}. \qquad \text{(by Lem. 165)}
\end{aligned}
$$

- $p_1 = 0 \wedge p_2 = 0$.
  By Lem. 12 we know $(\eta_1 \oplus_p \eta_2)^{(Stmt)}(\mathbf{skip}) = ({\eta_1}^{(Stmt)} \oplus_p {\eta_2}^{(Stmt)})(\mathbf{skip}) = p \cdot {\eta_1}^{(Stmt)}(\mathbf{skip}) + (1-p) \cdot {\eta_2}^{(Stmt)}(\mathbf{skip}) = p \cdot p_1 + (1-p) \cdot p_2 = 0$. By Lem. 287 we know $(\eta_1 \oplus_p \eta_2)|_{\mathbf{skip}} = \text{undefined}$.

Lemma 289. *For all* $\eta, R, \eta', \mathbf{q}$*, if* $\eta \overset{R}{\mapsto} \eta'$*,* $\eta^{(State)} \models \lceil \mathbf{q} \rceil$ *and* $\mathbf{sta}(\mathbf{q}, R)$*, then* $\eta'^{(State)} \models \lceil \mathbf{q} \rceil$.

Proof. For all $\eta, R, \eta', \mathbf{q}$ such that $\eta \overset{R}{\mapsto} \eta'$, $\eta^{(State)} \models \lceil \mathbf{q} \rceil$ and $\mathbf{sta}(\mathbf{q}, R)$, from $\eta \overset{R}{\mapsto} \eta'$ by Lem. 178 we know $\eta^{(State)} \overset{R}{\mapsto} \eta'^{(State)}$, so there exists θ such that $dom(\theta) = \eta^{(State)}$, $range(\theta) = \eta'^{(State)}$ and $\theta \subseteq [\![R]\!]$. To prove $\eta'^{(State)} \models \lceil \mathbf{q} \rceil$, we need to prove for all $\sigma' \in supp(\eta'^{(State)})$, $\sigma \models \mathbf{q}$. For all $\sigma' \in supp(\eta'^{(State)})$, from $range(\theta) = \eta'^{(State)}$ we have $\sigma' \in range(\theta)$, thus there exists σ such that $(\sigma, \sigma') \in \theta$, so $\sigma \in supp(\theta)$. From $dom(\theta) = \eta^{(State)}$ we know $\sigma \in \eta^{(State)}$. From $\eta^{(State)} \models \lceil \mathbf{q} \rceil$ we have $\sigma \models \mathbf{q}$. From $(\sigma, \sigma') \in \theta$ and $\theta \subseteq [\![R]\!]$ we know $(\sigma, \sigma') \models R$. From $\sigma \models \mathbf{q}$ and $\mathbf{sta}(\mathbf{q}, R)$ we have $\sigma' \models \mathbf{q}$.

Lemma 290. *For all* Q, η, η'*, if* $\mathbf{disablesplit}(Q, \eta)$ *and* $supp(\eta'^{(Stmt)}) \subseteq supp(\eta^{(Stmt)})$*, then* $\mathbf{disablesplit}(Q, \eta')$.

Proof. For all Q, η, η' such that $\mathbf{disablesplit}(Q, \eta)$ and $supp(\eta'^{(Stmt)}) \subseteq supp(\eta^{(Stmt)})$, to prove $\mathbf{disablesplit}(Q, \eta')$, we need to prove $\mathbf{disablesplit}(Q, C)$ for all $C \in supp(\eta'^{(Stmt)})$. For all $C \in supp(\eta'^{(Stmt)})$, from $supp(\eta'^{(Stmt)}) \subseteq supp(\eta^{(Stmt)})$ we know $C \in supp(\eta^{(Stmt)})$. From $\mathbf{disablesplit}(Q, \eta)$ we know $\mathbf{disablesplit}(Q, C)$.

Lemma 291. *For all* Q, η, θ, η'*, if* $\mathbf{disablesplit}(Q, \eta)$ *and* $\eta^{(State)} \models Q$*, then* $\eta \rightsquigarrow (\theta, \eta')$ *if and only if* $\eta \hookrightarrow (\theta, \eta')$.

Proof. For all Q, η, θ, η' such that $\mathbf{disablesplit}(Q, \eta)$ and $\eta^{(State)} \models Q$, there are two cases: $nextsplit(\eta) = \{\mathbf{split}(b_1, \ldots, b_k)\}$ or $\#nextsplit(\eta) > 1$. we prove the two cases respectively.

- $nextsplit(\eta) = \{\mathbf{split}(b_1, \ldots, b_k)\}$.
  There exists C and σ such that $(C, \sigma) \in supp(\eta)$ and $nextsplit(C) = \{\mathbf{split}(b_1, \ldots, b_k)\}$. From $(C, \sigma) \in supp(\eta)$ we know $C \in dom(supp(\eta))$, by Lem. 21 we know $dom(supp(\eta)) = supp(\eta^{(Stmt)})$, so $C \in supp(\eta^{(Stmt)})$. From $\mathbf{disablesplit}(Q, \eta)$ we know $\mathbf{disablesplit}(Q, C)$. From $nextsplit(C) = \{\mathbf{split}(b_1, \ldots, b_k)\}$ we know $\mathbf{disablesplit}(Q, \mathbf{split}(b_1, \ldots, b_k))$, thus there exists i such that $Q \Rightarrow \lceil b_i \rceil$. From $\eta'^{(State)} \models Q$ we know $\eta'^{(State)} \models \lceil b_i \rceil$. To prove $\eta \rightsquigarrow (\theta, \eta')$ if and only if $\eta \hookrightarrow (\theta, \eta')$, we prove the two directions respectively.
  - if $\eta \rightsquigarrow \eta'$, from $\eta'^{(State)} \models Q$ By Lem. 171 we know $\eta'|_{\text{true}} = \eta'$. From $\eta \rightsquigarrow \eta'$, $nextsplit(\eta) = \{\mathbf{split}(\text{true})\}$ and $\eta'|_{\text{true}} = \eta'$ we have $\eta \hookrightarrow \eta'$.
  - $nextsplit(\eta) \supset \{\mathbf{split}(\text{true})\}$.
    $\#nextsplit(\eta) > 1$, so $\eta \overset{t}{\hookrightarrow} \eta'$.
- if $\eta \hookrightarrow \eta'$, there are two cases.
  - case 1: there exists $\eta'', b_1, \ldots, b_k, i$ such that $\eta \rightsquigarrow \eta''$, $nextsplit(\eta) = \{\mathbf{split}(b_1, \ldots, b_k)\}$ and $\eta''|_{b_i} = \eta'$.
    From $nextsplit(\eta) \supseteq \{\mathbf{split}(\text{true})\}$ we know $k = i = 1$, $b_1 = \text{true}$. By Lem. 171 we know $\eta''|_{\text{true}} = \eta''$, so $\eta' = \eta''|_{b_i} = \eta''|_{\text{true}} = \eta''$. From $\eta \rightsquigarrow \eta''$ we have $\eta \rightsquigarrow \eta'$.

- case 2: $\#nextsplit(\eta) > 1$ and $\eta \rightsquigarrow \eta'$. trivial.

Lemma 292. *For all* $Q, C, \sigma, p, C', \sigma'$, *if* $\mathbf{disablesplit}(Q, C)$ *and* $(C, \sigma) \xrightarrow{p} (C', \sigma')$, *then* $\mathbf{disablesplit}(Q, C')$.

Proof. For all $Q, C, \sigma, p, C', \sigma'$ such that $\mathbf{disablesplit}(Q, C)$ and $(C, \sigma) \xrightarrow{p} (C', \sigma')$, we prove $\mathbf{disablesplit}(Q, C')$ by induction on the derivation of $(C, \sigma) \xrightarrow{p} (C', \sigma')$.

- case 1: $C = C' = \mathbf{skip}, \sigma = \sigma', p = 1$.
  From $\mathbf{disablesplit}(Q, \mathbf{skip})$ we know $\mathbf{disablesplit}(Q, C')$.
- case 2: $C = x := e, C' = \mathbf{skip}, \sigma' = \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}, p = 1$.
  From $\mathbf{disablesplit}(Q, \mathbf{skip})$ we know $\mathbf{disablesplit}(Q, C')$.
- case 3: $C = \mathbf{skip}; C_2, C' = C_2, \sigma = \sigma', p = 1$.
  From $\mathbf{disablesplit}(Q, C)$ we know $\mathbf{disablesplit}(Q, C_2)$, i.e., $\mathbf{disablesplit}(Q, C')$.
- case 4: $C = C_1; C_2, C_1 \neq \mathbf{skip}, C' = C_1'; C_2, (C_1, \sigma) \xrightarrow{p} (C_1', \sigma')$.
  IH: if $\mathbf{disablesplit}(Q, C_1)$ then $\mathbf{disablesplit}(Q, C_1')$.
  From $\mathbf{disablesplit}(Q, C)$ we know $\mathbf{disablesplit}(Q, C_1)$ and $\mathbf{disablesplit}(Q, C_2)$.
  From $\mathbf{disablesplit}(Q, C_1)$ by IH we have $\mathbf{disablesplit}(Q, C_1')$. From $\mathbf{disablesplit}(Q, C_2)$ we have $\mathbf{disablesplit}(Q, C_1'; C_2)$, i.e., $\mathbf{disablesplit}(Q, C')$.
- case 5: $C = \mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2, [\![b]\!]_\sigma = \mathrm{tt}, C' = C_1, \sigma' = \sigma, p = 1$.
  From $\mathbf{disablesplit}(Q, C)$ we know $\mathbf{disablesplit}(Q, C_1)$, i.e., $\mathbf{disablesplit}(Q, C')$.
- case 6: $C = \mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2, [\![b]\!]_\sigma = \mathrm{ff}, C' = C_2, \sigma' = \sigma, p = 1$.
  From $\mathbf{disablesplit}(Q, C)$ we know $\mathbf{disablesplit}(Q, C_2)$, i.e., $\mathbf{disablesplit}(Q, C')$.
- case 7: $C = \mathbf{while}\ (b)\ \mathbf{do}\ C_1, [\![b]\!]_\sigma = \mathrm{tt}, C' = C_1; \mathbf{while}\ (b)\ \mathbf{do}\ C_1, \sigma' = \sigma, p = 1$.
  From $\mathbf{disablesplit}(Q, C)$ we know $\mathbf{disablesplit}(Q, C_1)$, thus $\mathbf{disablesplit}(Q, C_1; \mathbf{while}\ (b)\ \mathbf{do}\ C_1)$, i.e., $\mathbf{disablesplit}(Q, C')$.
- case 8: $C = \mathbf{while}\ (b)\ \mathbf{do}\ C_1, [\![b]\!]_\sigma = \mathrm{ff}, C' = \mathbf{skip}, \sigma' = \sigma, p = 1$.
  From $\mathbf{disablesplit}(Q, \mathbf{skip})$ we know $\mathbf{disablesplit}(Q, C')$.
- case 9: $C = \langle C_1 \rangle, C' = \mathbf{skip}$.
  From $\mathbf{disablesplit}(Q, \mathbf{skip})$ we know $\mathbf{disablesplit}(Q, C')$.
- case 10: $C = \langle C_1 \rangle\ sp, C' = \mathbf{skip}, (\langle C_1 \rangle, \sigma) \xrightarrow{p} (\mathbf{skip}, \sigma')$.
  From $\mathbf{disablesplit}(Q, \mathbf{skip})$ we know $\mathbf{disablesplit}(Q, C')$.
- case 11: $C = \langle C_1 \rangle \oplus_{p'} \langle C_2 \rangle, C' = \langle C_1 \rangle, \sigma = \sigma', p = p'$.
  From $\mathbf{disablesplit}(Q, \langle C_1 \rangle)$ we know $\mathbf{disablesplit}(Q, C')$.
- case 12: $C = \langle C_1 \rangle \oplus_{p'} \langle C_2 \rangle, C' = \langle C_2 \rangle, \sigma = \sigma', p = 1 - p'$.
  From $\mathbf{disablesplit}(Q, \langle C_2 \rangle)$ we know $\mathbf{disablesplit}(Q, C')$.

Lemma 293. *For all* Q, η, θ, η', *if* $\mathbf{disablesplit}(Q, \eta)$ *and* $\eta \rightsquigarrow (\theta, \eta')$, *then* $\mathbf{disablesplit}(Q, \eta')$.

Proof. For all Q, η, θ, η' such that $\mathbf{disablesplit}(Q, \eta)$ and $\eta \rightsquigarrow (\theta, \eta')$, to prove $\mathbf{disablesplit}(Q, \eta')$, we need to prove $\mathbf{disablesplit}(Q, C')$ for all $C' \in supp(\eta'^{(Stmt)})$. By Lem. 21 we know $supp(\eta'^{(Stmt)}) = dom(supp(\eta'))$. For all $C' \in supp(\eta'^{(Stmt)})$, we have $C' \in dom(supp(\eta'))$, so there exists σ' such that $(C', \sigma') \in supp(\eta')$, i.e., $\eta'(C', \sigma') > 0$. From $\eta \rightsquigarrow (\theta, \eta')$ we know $\eta'(C', \sigma') = \sum_{C,\sigma} \{\eta(C, \sigma) \cdot p \mid$

$(C,\sigma) \xrightarrow{p} (C',\sigma')\} > 0$, thus there exists C and σ such that $\eta(C,\sigma) > 0$, $p > 0$ and $(C,\sigma) \xrightarrow{p} (C',\sigma')$. From $\eta(C,\sigma) > 0$ we know $(C,\sigma) \in \mathit{supp}(\eta)$, so $C \in \mathit{dom}(\mathit{supp}(\eta))$. By Lem. 21 we know $\mathit{dom}(\mathit{supp}(\eta)) = \mathit{supp}(\eta^{(Stmt)})$, thus $C \in \mathit{supp}(\eta^{(Stmt)})$. From $\mathbf{disablesplit}(Q,\eta)$ we know $\mathbf{disablesplit}(Q,C)$. From $(C,\sigma) \xrightarrow{p} (C',\sigma')$ by Lem. 292 we have $\mathbf{disablesplit}(Q,C')$.

Lemma 294. *For all* Q,η,C_2*, if* $\mathbf{disablesplit}(Q,\eta)$ *and* $\mathbf{disablesplit}(Q,C_2)$*, then* $\mathbf{disablesplit}(Q,\eta;C_2)$.

Proof. For all Q,η,C_2 such that $\mathbf{disablesplit}(Q,\eta)$ and $\mathbf{disablesplit}(C_2)$, to prove $\mathbf{disablesplit}(Q,\eta;C_2)$, we need to prove $\mathbf{disablesplit}(Q,C)$ for all $C \in \mathit{supp}(\eta;{C_2}^{(Stmt)})$. By Lem. 21 we know $\mathit{supp}(\eta;{C_2}^{(Stmt)}) = \mathit{dom}(\mathit{supp}(\eta;C_2))$ and $\mathit{supp}(\eta^{(Stmt)}) = \mathit{dom}(\mathit{supp}(\eta))$. For all $C \in \mathit{supp}(\eta;{C_2}^{(Stmt)})$, we have $C \in \mathit{dom}(\mathit{supp}(\eta;C_2))$, so there exists σ such that $(C,\sigma) \in \mathit{supp}(\eta;C_2)$, i.e., $\eta;C_2(C,\sigma) > 0$, thus there exists C_1 such that $C = C_1;C_2$ and $\eta(C_1,\sigma) > 0$, i.e., $(C_1,\sigma) \in \mathit{supp}(\eta)$, so $C_1 \in \mathit{dom}(\mathit{supp}(\eta))$. From $\mathit{supp}(\eta^{(Stmt)}) = \mathit{dom}(\mathit{supp}(\eta))$ we know $C_1 \in \mathit{supp}(\eta^{(Stmt)})$. From $\mathbf{disablesplit}(Q,\eta)$ we know $\mathbf{disablesplit}(Q,C_1)$. From $\mathbf{disablesplit}(Q,C_2)$ we have $\mathbf{disablesplit}(Q,C_1;C_2)$, i.e., $\mathbf{disablesplit}(Q,C)$.

Lemma 295. *For all* Q,η_1,η_2,p*, if* $\mathbf{disablesplit}(Q,\eta_1)$ *and* $\mathbf{disablesplit}(Q,\eta_2)$*, then* $\mathbf{disablesplit}(Q,\eta_1 \oplus_p \eta_2)$.

Proof. For all Q,η_1,η_2,p such that $\mathbf{disablesplit}(Q,\eta_1)$ and $\mathbf{disablesplit}(Q,\eta_2)$, there are three cases: $p = 0$, $p = 1$ or $0 < p < 1$. We prove the three cases respectively.

- $p = 0$.
  $\eta_1 \oplus_p \eta_2 = \eta_1 \oplus_0 \eta_2 = \eta_2$. From $\mathbf{disablesplit}(Q,\eta_2)$ we know $\mathbf{disablesplit}(Q,\eta_1 \oplus_p \eta_2)$.
- $p = 1$.
  $\eta_1 \oplus_p \eta_2 = \eta_1 \oplus_1 \eta_2 = \eta_1$. From $\mathbf{disablesplit}(Q,\eta_1)$ we know $\mathbf{disablesplit}(Q,\eta_1 \oplus_p \eta_2)$.
- $0 < p < 1$.
  To prove $\mathbf{disablesplit}(Q,\eta_1 \oplus_p \eta_2)$, we need to prove $\mathbf{disablesplit}(Q,C)$ for all $C \in \mathit{supp}((\eta_1 \oplus_p \eta_2)^{(Stmt)})$. From $0 < p < 1$ by Lem. 11 and Lem. 275 we know $\mathit{supp}((\eta_1 \oplus_p \eta_2)^{(Stmt)}) = \mathit{supp}({\eta_1}^{(Stmt)} \oplus_p {\eta_2}^{(Stmt)}) = \mathit{supp}({\eta_1}^{(Stmt)}) \cup \mathit{supp}({\eta_1}^{(Stmt)})$. For all $C \in \mathit{supp}((\eta_1 \oplus_p \eta_2)^{(Stmt)})$, we have $C \in \mathit{supp}({\eta_1}^{(Stmt)}) \cup \mathit{supp}({\eta_1}^{(Stmt)})$. There are two cases: $C \in \mathit{supp}({\eta_1}^{(Stmt)})$ or $C \in \mathit{supp}({\eta_2}^{(Stmt)})$.
  If $C \in \mathit{supp}({\eta_1}^{(Stmt)})$, from $\mathbf{disablesplit}(Q,\eta_1)$ we know $\mathbf{disablesplit}(Q,C)$.
  If $C \in \mathit{supp}({\eta_2}^{(Stmt)})$, from $\mathbf{disablesplit}(Q,\eta_2)$ we know $\mathbf{disablesplit}(Q,C)$.

Lemma 296. *For all* η*, there exists* θ *and* η' *such that* $\eta \rightsquigarrow (\theta,\eta')$.

Proof. For all η, let $\eta' \stackrel{\text{def}}{=} \lambda(C',\sigma').\sum_{C,\sigma}\{\eta(C,\sigma) \cdot p \mid (C,\sigma) \xrightarrow{p} (C',\sigma')\}$ and $\theta = \{(\sigma,\sigma') \mid \exists C,C'.\ \eta(C,\sigma) > 0 \wedge (C,\sigma) \xrightarrow{p} (C',\sigma') \wedge p > 0\}$, then $\eta \rightsquigarrow (\theta,\eta')$.

Lemma 297. *For all* $R, G, \mathbf{q}, Q, n, \eta_1, \eta_2, p$, *if* $0 < p < 1$, $(\eta_1 \oplus_p \eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^n_{NST} (G, Q)$, $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{scl}(Q)$, $\mathbf{Id} \Rightarrow R$ *and* $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1 \oplus_p \eta_2)$, *then* $(\eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^n_{NST} (G, Q)$.

Proof. For all $R, G, \mathbf{q}, Q, n$ such that $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{scl}(Q)$ and $\mathbf{Id} \Rightarrow R$, we prove for all η_1, η_2, p, if $0 < p < 1$, $(\eta_1 \oplus_p \eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^n_{\text{NST}} (G, Q)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1 \oplus_p \eta_2)$, then $(\eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^n_{\text{NST}} (G, Q)$ by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all η_1, η_2, p, if $0 < p < 1$, $(\eta_1 \oplus_p \eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G, Q)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1 \oplus_p \eta_2)$, then $(\eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G, Q)$.
  For all η_1, η_2, p such that $0 < p < 1$, $(\eta_1 \oplus_p \eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1 \oplus_p \eta_2)$, from $0 < p < 1$ by Lem. 275 we know $supp(\eta_1 \oplus_p \eta_2) = supp(\eta_1) \cup supp(\eta_2)$, thus $supp(\eta_1) \subseteq supp(\eta)$ and $supp(\eta_2) \subseteq supp(\eta)$. To prove $(\eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$, we need to prove
  - if ${\eta_2}^{(Stmt)}(\mathbf{skip}) > 0$, then ${\eta_2|_{\mathbf{skip}}}^{(State)} \models Q$.
    From $p < 1$ we have $1 - p > 0$. By Lem. 12 we know ${(\eta_1 \oplus_p \eta_2)}^{(Stmt)}(\mathbf{skip}) = ({\eta_1}^{(Stmt)} \oplus_p {\eta_2}^{(Stmt)})(\mathbf{skip}) = p \cdot {\eta_1}^{(Stmt)}(\mathbf{skip}) + (1-p) \cdot {\eta_2}^{(Stmt)}(\mathbf{skip}) > 0$. From $(\eta_1 \oplus_p \eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^n_{\text{NST}} (G, Q)$ we have ${(\eta_1 \oplus_p \eta_2)|_{\mathbf{skip}}}^{(State)} \models Q$. From $supp(\eta_2) \subseteq supp(\eta_1 \oplus_p \eta_2)$, ${\eta_2}^{(Stmt)}(\mathbf{skip}) > 0$ and ${(\eta_1 \oplus_p \eta_2)}^{(Stmt)}(\mathbf{skip}) > 0$ by Lem. 277 we know $supp(\eta_2|_{\mathbf{skip}}) \subseteq supp((\eta_1 \oplus_p \eta_2)|_{\mathbf{skip}})$. By Lem. 24 we have $supp({\eta_2|_{\mathbf{skip}}}^{(State)}) \subseteq supp({(\eta_1 \oplus_p \eta_2)|_{\mathbf{skip}}}^{(State)})$. From ${\eta_2|_{\mathbf{skip}}}^{(State)} \models Q$ and $\mathbf{scl}(Q)$ we have ${\eta_2|_{\mathbf{skip}}}^{(State)} \models Q$.
  - ${\eta_2}^{(State)} \models \lceil \mathbf{q} \rceil$.
    From $(\eta_1 \oplus_p \eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^n_{\text{NST}} (G, Q)$ we know ${(\eta_1 \oplus_p \eta_2)}^{(State)} \models \lceil \mathbf{q} \rceil$. From $supp(\eta_2) \subseteq supp(\eta_1 \oplus_p \eta_2)$ by Lem. 24 we know $supp({\eta_2}^{(State)}) \subseteq supp({(\eta_1 \oplus_p \eta_2)}^{(State)})$.
    From ${(\eta_1 \oplus_p \eta_2)}^{(State)} \models \lceil \mathbf{q} \rceil$ and $\mathbf{scl}(\lceil \mathbf{q} \rceil)$ we have ${\eta_2}^{(State)} \models \lceil \mathbf{q} \rceil$.
  - for all η_2', if $\eta_2 \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta_2'$, then $(\eta_2', R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G, Q)$. For all η_2' such that $\eta_2 \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta_2'$, there exists η_2'' and b such that $\eta \overset{R}{\mapsto} \eta_2''$, $\eta_2''|_b = \eta_2'$ and ${\eta_2'}^{(State)} \models \lceil \mathbf{q} \rceil$. From $\mathbf{Id} \Rightarrow R$ by Lem. 273 we know $\eta_1 \overset{R}{\mapsto} \eta_1$. From $0 < p < 1$ and $\eta \overset{R}{\mapsto} \eta_2''$ by Lem. 236 we have $\eta_1 \oplus_p \eta_2 \overset{R}{\mapsto} \eta_1 \oplus_p \eta_2''$.
    Let $p_1 \overset{\text{def}}{=} [\![\mathbf{Pr}(b)]\!]_{{\eta_1}^{(State)}}$ and $p_2 \overset{\text{def}}{=} [\![\mathbf{Pr}(b)]\!]_{{\eta_2''}^{(State)}}$. From $\eta_2''|_b = \eta_2'$ by Lem. 205 we know $p_2 = [\![\mathbf{Pr}(b)]\!]_{{\eta_2''}^{(State)}} > 0$. There are two cases: $p_1 > 0$ or $p_1 = 0$. We prove the two cases respectively.
    - $p_1 > 0$.
      Let $p' \overset{\text{def}}{=} \frac{p \cdot p_1}{p \cdot p_1 + (1-p) \cdot p_2}$. From $0 < p < 1$, $p_1 > 0$ and $p_2 > 0$ we know $p' > 0$. From $0 < p < 1$, $[\![\mathbf{Pr}(b)]\!]_{{\eta_1}^{(State)}} = p_1 > 0$ and $[\![\mathbf{Pr}(b)]\!]_{{\eta_2''}^{(State)}} = p_2 > 0$ by Lem. 238 we know $(\eta_1 \oplus_p \eta_2'')|_b = \eta_1|_b \oplus_{p'} \eta_2''|_b = \eta_1|_b \oplus_{p'} \eta_2'$. By Lem. 20 we know $supp(\eta_1|_b) \subseteq supp(\eta_1) \subseteq$

$supp(\eta_1 \oplus_p \eta_2)$. By Lem. 24 we know $supp(\eta_1|_b{}^{(State)}) \subseteq supp((\eta_1 \oplus_p \eta_2)^{(State)})$. From $(\eta_1 \oplus_p \eta_2)^{(State)} \models \lceil \mathbf{q} \rceil$ and $\mathbf{scl}(\lceil \mathbf{q} \rceil)$ we know $\eta_1|_b{}^{(State)} \models \lceil \mathbf{q} \rceil$. From $\eta_2'{}^{(State)} \models \lceil \mathbf{q} \rceil$ and $\mathbf{closed}(\lceil \mathbf{q} \rceil)$ we know $\eta_1|_b{}^{(State)} \oplus_{p'} \eta_2'{}^{(State)} \models \lceil \mathbf{q} \rceil$. By Lem. 12 we know $(\eta_1|_b \oplus_{p'} \eta_2')^{(State)} = \eta_1|_b{}^{(State)} \oplus_{p'} \eta_2'{}^{(State)} \models \lceil \mathbf{q} \rceil$. From $\eta_1 \oplus_p \eta_2 \overset{R}{\mapsto} \eta_1 \oplus_p \eta_2''$, $(\eta_1 \oplus_p \eta_2'')|_b = \eta_1|_b \oplus_{p'} \eta_2'$ and $(\eta_1|_b \oplus_{p'} \eta_2')^{(State)} \models \lceil \mathbf{q} \rceil$ we know $\eta_1 \oplus_p \eta_2 \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta_1|_b \oplus_{p'} \eta_2'$. From $(\eta_1 \oplus_p \eta_2, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k+1}_{\text{NST}} (G, Q)$ we know $(\eta_1|_b \oplus_{p'} \eta_2', R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k}_{\text{NST}} (G, Q)$. From $\eta_1 \oplus_p \eta_2 \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta_1|_b \oplus_{p'} \eta_2'$ by Lem. 188 we know $supp((\eta_1|_b \oplus_{p'} \eta_2')^{(Stmt)}) \subseteq supp((\eta_1 \oplus_p \eta_2)^{(Stmt)})$. From $\mathbf{disablesplit}(\eta_1 \oplus_p \eta_2)$ by Lem. 290 we have $\mathbf{disablesplit}(\eta_1|_b \oplus_{p'} \eta_2')$. From $0 < p' < 1$ and $(\eta_1|_b \oplus_{p'} \eta_2', R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k}_{\text{NST}} (G, Q)$ by IH we have $(\eta_2', R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k}_{\text{NST}} (G, Q)$.

  * $p_1 = 0$.
    From $0 < p < 1$, $[\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}} = p_1 = 0$ and $[\![\mathbf{Pr}(b)]\!]_{\eta_2''{}^{(State)}} = p_2 > 0$ by Lem. 238 we know $(\eta_1 \oplus_p \eta_2'')|_b = \eta_2''|_b = \eta_2'$. From $\eta_1 \oplus_p \eta_2 \overset{R}{\mapsto} \eta_1 \oplus_p \eta_2''$, $(\eta_1 \oplus_p \eta_2'')|_b = \eta_2'$ and $\eta_2'{}^{(State)} \models \lceil \mathbf{q} \rceil$ we know $\eta_1 \oplus_p \eta_2 \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta_2'$. From $(\eta_1 \oplus_p \eta_2, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k+1}_{\text{NST}} (G, Q)$ we know $(\eta_2', R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k}_{\text{NST}} (G, Q)$.

- for all θ_2, η_2', if $\eta_2 \hookrightarrow (\theta_2, \eta_2')$, then $\theta_2 \subseteq [\![G]\!]$, $\eta_2'{}^{(State)} \models \lceil \mathbf{q} \rceil$ and $(\eta_2', R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k}_{\text{NST}} (G, Q)$.
  For all θ_2, η_2' such that $\eta_2 \hookrightarrow (\theta_2, \eta_2')$, from $supp(\eta_2) \subseteq supp(\eta_1 \oplus_p \eta_2)$ by Lem. 23 we know $supp(\eta_2{}^{(Stmt)}) \subseteq supp((\eta_1 \oplus_p \eta_2)^{(Stmt)})$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1 \oplus_p \eta_2)$ by Lem. 290 we know $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_2)$. From $\eta_2 \hookrightarrow (\theta_2, \eta_2')$ and $\eta_2{}^{(State)} \models \lceil \mathbf{q} \rceil$ by Lem. 291 we know $\eta_2 \rightsquigarrow (\theta_2, \eta_2')$. By Lem. 296 there exists θ_1 and η_1' such that $\eta_1 \rightsquigarrow (\theta_1, \eta_1')$. From $0 < p < 1$ and $\eta_2 \rightsquigarrow (\theta_2, \eta_2')$ by Lem. 246 we know $\eta_1 \oplus_p \eta_2 \rightsquigarrow (\theta_1 \cup \theta_2, \eta_1' \oplus_p \eta_2')$. From $\mathbf{disablesplit}(\eta_1 \oplus_p \eta_2)$ and $(\eta_1 \oplus_p \eta_2)^{(State)} \models \lceil \mathbf{q} \rceil$ by Lem. 291 we know $\eta_1 \oplus_p \eta_2 \hookrightarrow (\theta_1 \cup \theta_2, \eta_1' \oplus_p \eta_2')$. From $(\eta_1 \oplus_p \eta_2, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k+1}_{\text{NST}} (G, Q)$ we know $\theta_1 \cup \theta_2 \subseteq [\![G]\!]$, $(\eta_1' \oplus_p \eta_2')^{(State)} \models \lceil \mathbf{q} \rceil$ and $(\eta_1' \oplus_p \eta_2', R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k}_{\text{NST}} (G, Q)$, thus $\theta_2 \subseteq \theta_1 \cup \theta_2 \subseteq [\![G]\!]$. From $0 < p < 1$ by Lem. 275 we know $supp(\eta_1' \oplus_p \eta_2') = supp(\eta_1') \cup supp(\eta_2') \supseteq supp(\eta_2')$. By Lem. 24 we know $supp(\eta_2'{}^{(State)}) \subseteq supp((\eta_1' \oplus_p \eta_2')^{(State)})$. From $(\eta_1' \oplus_p \eta_2')^{(State)} \models \lceil \mathbf{q} \rceil$ and $\mathbf{scl}(\lceil \mathbf{q} \rceil)$ we have $\eta_2'{}^{(State)} \models \lceil \mathbf{q} \rceil$. From $\mathbf{disablesplit}(\eta_1 \oplus_p \eta_2)$ and $\eta_1 \oplus_p \eta_2 \rightsquigarrow (\theta_1 \cup \theta_2, \eta_1' \oplus_p \eta_2')$ by Lem. 293 we know $\mathbf{disablesplit}(\eta_1' \oplus_p \eta_2')$. From $(\eta_1' \oplus_p \eta_2') \Longrightarrow^{k}_{\text{NST}} (G, Q)$ by IH we have $(\eta_2', R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k}_{\text{NST}} (G, Q)$.

Lemma 298. *For all* $R, G, Q, \mathbf{q}, n, \eta_1, \eta_2, p$, *if* $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{n}_{NST} (G, Q)$, $(\eta_2, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{n}_{NST} (G, Q)$, $\mathbf{closed}(Q)$, $0 < p < 1$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$ *and* $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_2)$, *then* $(\eta_1 \oplus_p \eta_2, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{n}_{NST} (G, Q)$.

Proof. For all $R, G, Q, \mathbf{q}, n$ such that $\mathbf{closed}(Q)$, we prove for all η_1, η_2, p, if $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{n}_{\text{NST}} (G, Q)$, $(\eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{n}_{\text{NST}} (G, Q)$, $0 < p < 1$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_2)$, then $(\eta_1 \oplus_p \eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{n}_{\text{NST}} (G, Q)$ by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all η_1, η_2, p, if $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$, $(\eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$, $0 < p < 1$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_2)$, then $(\eta_1 \oplus_p \eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$.
  For all η_1, η_2, p such that $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$, $(\eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$, $0 < p < 1$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_2)$, to prove $(\eta_1 \oplus_p \eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$, we need to prove
  - if $(\eta_1 \oplus_p \eta_2)^{(Stmt)}(\mathbf{skip}) > 0$, then $(\eta_1 \oplus_p \eta_2)|_{\mathbf{skip}}{}^{(State)} \models Q$.
    Let $p_1 \stackrel{\text{def}}{=} \eta_1{}^{(Stmt)}(\mathbf{skip})$ and $p_2 \stackrel{\text{def}}{=} \eta_2{}^{(Stmt)}(\mathbf{skip})$. By Lem. 12 we know $(\eta_1 \oplus_p \eta_2)^{(Stmt)}(\mathbf{skip}) = (\eta_1{}^{(Stmt)} \oplus_p \eta_2{}^{(Stmt)})(\mathbf{skip}) = p \cdot \eta_1{}^{(Stmt)}(\mathbf{skip}) + (1-p) \cdot \eta_2{}^{(Stmt)}(\mathbf{skip}) = p \cdot p_1 + (1-p) \cdot p_2$. From $(\eta_1 \oplus_p \eta_2)^{(Stmt)}(\mathbf{skip}) > 0$ we know there are three cases: $p_1 > 0 \wedge p_2 > 0$, $p_1 > 0 \wedge p_2 = 0$, or $p_1 = 0 \wedge p_2 > 0$. We prove the three cases respectively.
    - $p_1 > 0 \wedge p_2 > 0$.
      From $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ and $\eta_1{}^{(Stmt)}(\mathbf{skip}) = p_1 > 0$ we know $\eta_1|_{\mathbf{skip}}{}^{(State)} \models Q$. From $(\eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ and $\eta_2{}^{(Stmt)}(\mathbf{skip}) = p_2 > 0$ we know $\eta_2|_{\mathbf{skip}}{}^{(State)} \models Q$. Let $p' \stackrel{\text{def}}{=} \frac{p \cdot p_1}{p \cdot p_1 + (1-p) \cdot p_2}$. From $0 < p < 1$, $p_1 > 0$ and $p_2 > 0$ we know $0 < p' < 1$. From $0 < p < 1$, $\eta_1{}^{(Stmt)}(\mathbf{skip}) = p_1 > 0$ and $\eta_2{}^{(Stmt)}(\mathbf{skip}) = p_2 > 0$ by Lem. 234 we know $(\eta_1 \oplus_p \eta_2)|_{\mathbf{skip}} = \eta_1|_{\mathbf{skip}} \oplus_{p'} \eta_2|_{\mathbf{skip}}$. From $\eta_1|_{\mathbf{skip}}{}^{(State)} \models Q$, $\eta_2|_{\mathbf{skip}}{}^{(State)} \models Q$ and $\mathbf{closed}(Q)$ we have $\eta_1|_{\mathbf{skip}}{}^{(State)} \oplus_{p'} \eta_2|_{\mathbf{skip}}{}^{(State)} \models Q$.
    - $p_1 > 0 \wedge p_2 = 0$.
      From $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ and $\eta_1{}^{(Stmt)}(\mathbf{skip}) = p_1 > 0$ we know $\eta_1|_{\mathbf{skip}}{}^{(State)} \models Q$. From $0 < p < 1$, $\eta_1{}^{(Stmt)}(\mathbf{skip}) = p_1 > 0$ and $\eta_2{}^{(Stmt)}(\mathbf{skip}) = p_2 = 0$ by Lem. 234 we know $(\eta_1 \oplus_p \eta_2)|_{\mathbf{skip}} = \eta_1|_{\mathbf{skip}} \models Q$.
    - $p_1 = 0 \wedge p_2 > 0$.
      From $(\eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ and $\eta_2{}^{(Stmt)}(\mathbf{skip}) = p_2 > 0$ we know $\eta_2|_{\mathbf{skip}}{}^{(State)} \models Q$. From $0 < p < 1$, $\eta_1{}^{(Stmt)}(\mathbf{skip}) = p_1 = 0$ and $\eta_2{}^{(Stmt)}(\mathbf{skip}) = p_2 > 0$ by Lem. 234 we know $(\eta_1 \oplus_p \eta_2)|_{\mathbf{skip}} = \eta_2|_{\mathbf{skip}} \models Q$.
  - $(\eta_1 \oplus_p \eta_2)^{(State)} \models \lceil \mathbf{q} \rceil$.
    From $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ we know $\eta_1{}^{(State)} \models \lceil \mathbf{q} \rceil$. From $(\eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ we know $\eta_2{}^{(State)} \models \lceil \mathbf{q} \rceil$. From $\mathbf{closed}(\lceil \mathbf{q} \rceil)$ we have $\eta_1{}^{(State)} \oplus_p \eta_2{}^{(State)} \models \lceil \mathbf{q} \rceil$. By Lem. 12 we have $(\eta_1 \oplus_p \eta_2)^{(State)} = \eta_1{}^{(State)} \oplus_p \eta_2{}^{(State)} \models \lceil \mathbf{q} \rceil$.

- for all η', if $\eta_1 \oplus_p \eta_2 \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\square} (G, Q)$.

  For all η' such that $\eta_1 \oplus_p \eta_2 \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, there exists η'' and b such that $\eta_1 \oplus_p \eta_2 \overset{R}{\mapsto} \eta''$, $\eta''|_b = \eta'$ and $\eta'^{(State)} \models \lceil \mathbf{q} \rceil$. From $0 < p < 1$ and $\eta_1 \oplus_p \eta_2 \overset{R}{\mapsto} \eta''$ by Lem. 235 there exists η''_1, η''_2, p'' such that $0 < p'' < 1$, $\eta'' = \eta''_1 \oplus_{p''} \eta''_2$, $\eta_1 \overset{R}{\mapsto} \eta''_1$ and $\eta_2 \overset{R}{\mapsto} \eta''_2$. Let $p_1 \overset{\text{def}}{=} [\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}}$ and $p_2 \overset{\text{def}}{=} [\![\mathbf{Pr}(b)]\!]_{\eta_2{}^{(State)}}$. From $\eta''|_b = \eta'$ by Lem. 205 we know $[\![\mathbf{Pr}(b)]\!]_{\eta''{}^{(State)}} > 0$. By Lem. 12 and Lem. 237 we know $[\![\mathbf{Pr}(b)]\!]_{\eta''{}^{(State)}} = [\![\mathbf{Pr}(b)]\!]_{(\eta''_1 \oplus_p \eta''_2)^{(State)}} = [\![\mathbf{Pr}(b)]\!]_{\eta''_1{}^{(State)} \oplus_p \eta''_2{}^{(State)}} = p \cdot [\![\mathbf{Pr}(b)]\!]_{\eta''_1{}^{(State)}} + (1-p) \cdot [\![\mathbf{Pr}(b)]\!]_{\eta''_2{}^{(State)}} = p \cdot p_1 + (1-p) \cdot p_2 > 0$. There are three cases: $p_1 > 0 \wedge p_2 > 0$, $p_1 > 0 \wedge p_2 = 0$ or $p_2 = 0 \wedge p_2 > 0$. We prove the three cases respectively.

  * $p_1 > 0 \wedge p_2 > 0$.
    Let $p' \overset{\text{def}}{=} \frac{p'' \cdot p_1}{p'' \cdot p_1 + (1-p'') \cdot p_2}$. From $0 < p'' < 1$, $p_1 > 0$ and $p_2 > 0$ we know $0 < p' < 1$. From $0 < p'' < 1$, $[\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}} = p_1 > 0$ and $[\![\mathbf{Pr}(b)]\!]_{\eta_2{}^{(State)}} = p_2 > 0$ by Lem. 238 we know $\eta' = \eta''|_b = (\eta''_1 \oplus_{p''} \eta''_2)|_b = \eta''_1|_b \oplus_{p'} \eta''_2|_b$. By Lem. 12 we know $\eta'^{(State)} = \eta''_1|_b{}^{(State)} \oplus_{p'} \eta''_2|_b{}^{(State)}$. From $0 < p' < 1$ by Lem. 275 we know $supp(\eta'^{(State)}) = supp(\eta''_1|_b{}^{(State)}) \cup supp(\eta''_2|_b{}^{(State)}) \supseteq supp(\eta''_1|_b{}^{(State)})$. From $\eta'^{(State)} \models \lceil \mathbf{q} \rceil$ and $\mathbf{scl}(\lceil \mathbf{q} \rceil)$ we know $\eta''_1|_b{}^{(State)} \models \lceil \mathbf{q} \rceil$. From $\eta_1 \overset{R}{\mapsto} \eta''_1$ we have $\eta_1 \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta''_1|_b$. From $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ we know $(\eta''_1|_b, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G, Q)$. From $\eta_1 \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta''_1|_b$ by Lem. 188 we know $supp(\eta''_1|_b{}^{(Stmt)}) \subseteq supp(\eta_1{}^{(Stmt)})$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$ by Lem. 290 we have $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta''_1|_b)$. Similarly we can prove $(\eta''_2|_b, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G, Q)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta''_2|_b)$. From $(\eta''_1|_b, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G, Q)$, $(\eta''_2|_b, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G, Q)$, $0 < p'' < 1$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta''_1|_b)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta''_2|_b)$ by IH we have $(\eta''_1 \oplus_{p'} \eta''_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G, Q)$, i.e., $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G, Q)$.
  * $p_1 > 0 \wedge p_2 = 0$.
    From $0 < p'' < 1$, $[\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}} = p_1 > 0$ and $[\![\mathbf{Pr}(b)]\!]_{\eta_2{}^{(State)}} = p_2 = 0$ by Lem. 238 we know $\eta' = \eta''|_b = (\eta''_1 \oplus_{p''} \eta''_2)|_b = \eta''_1|_b$. From $\eta_1 \overset{R}{\mapsto} \eta''_1$, $\eta''_1|_b = \eta'$ and $\eta'^{(State)} \models \lceil \mathbf{q} \rceil$ we have $\eta_1 \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta'$. From $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ we know $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G, Q)$.
  * $p_1 = 0 \wedge p_2 > 0$.
    From $0 < p'' < 1$, $[\![\mathbf{Pr}(b)]\!]_{\eta_1{}^{(State)}} = p_1 = 0$ and $[\![\mathbf{Pr}(b)]\!]_{\eta_2{}^{(State)}} = p_2 > 0$ by Lem. 238 we know $\eta' = \eta''|_b = (\eta''_1 \oplus_{p''} \eta''_2)|_b = \eta''_2|_b$. From $\eta_2 \overset{R}{\mapsto} \eta''_2$, $\eta''_2|_b = \eta'$ and $\eta'^{(State)} \models \lceil \mathbf{q} \rceil$ we have $\eta_2 \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta'$. From $(\eta_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ we know $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G, Q)$.

- for all θ and η', if $(\eta_1 \oplus_p \eta_2) \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, ${\eta'}^{(State)} \models \lceil \mathbf{q} \rceil$ and $(\eta', R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\square} (G, Q)$.
  For all θ and η' such that $(\eta_1 \oplus_p \eta_2) \hookrightarrow (\theta, \eta')$, from $0 < p < 1$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_2)$ by Lem. 295 we have $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1 \oplus_p \eta_2)$. From ${(\eta_1 \oplus_p \eta_2)}^{(State)} \models \lceil \mathbf{q} \rceil$ and $(\eta_1 \oplus_p \eta_2) \hookrightarrow (\theta, \eta')$ we know $(\eta_1 \oplus_p \eta_2) \rightsquigarrow (\theta, \eta')$. From $0 < p < 1$ by Lem. 245 there exists $\theta_1, \theta_2, \eta'_1, \eta'_2$ such that $\theta = \theta_1 \cup \theta_2$, $\eta' = \eta'_1 \oplus_p \eta'_2$, $\eta_1 \rightsquigarrow (\theta_1, \eta'_1)$ and $\eta_2 \rightsquigarrow (\theta_2, \eta'_2)$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$, ${\eta_1}^{(State)} \models \lceil \mathbf{q} \rceil$ and $\eta_1 \rightsquigarrow (\theta_1, \eta'_1)$ by Lem. 291 we know $\eta_1 \hookrightarrow (\theta_1, \eta'_1)$. From $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k+1}_{\text{NST}} (G, Q)$ we know $\theta_1 \subseteq [\![G]\!]$, ${\eta'_1}^{(State)} \models \lceil \mathbf{q} \rceil$ and $(\eta'_1, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$. Similarly we can prove $\theta_2 \subseteq [\![G]\!]$, ${\eta'_2}^{(State)} \models \lceil \mathbf{q} \rceil$ and $(\eta'_2, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$. thus $\theta = \theta_1 \cup \theta_2 \subseteq [\![G]\!]$. By Lem. 12 we have ${\eta'}^{(State)} = {(\eta'_1 \oplus_p \eta'_2)}^{(State)} = {\eta'_1}^{(State)} \oplus_p {\eta'_2}^{(State)}$. From ${\eta'_1}^{(State)} \models \lceil \mathbf{q} \rceil$, ${\eta'_2}^{(State)} \models \lceil \mathbf{q} \rceil$ and $\mathbf{closed}(\lceil \mathbf{q} \rceil)$ we have ${\eta'}^{(State)} \models \lceil \mathbf{q} \rceil$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$ and $\eta_1 \rightsquigarrow (\theta_1, \eta'_1)$ by Lem. 293 we have $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta'_1)$. Similarly we can prove $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta'_2)$. From $(\eta'_1, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$, $(\eta'_2, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$, $0 < p < 1$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta'_1)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta'_2)$ by IH we have $(\eta'_1 \oplus_p \eta'_2, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$, i.e., $(\eta', R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$.

Lemma 299. *For all $R, G, I, P, Q, \mathbf{q}, C_2, n, \eta$, if $(\eta, R, \lceil \mathbf{q} \rceil) \Longrightarrow^n_{NST} (G, P)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C_2)$, $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{closed}(Q)$, $\mathbf{scl}(P)$, $\mathbf{Id} \Rightarrow R$, $\mathbf{Id} \Rightarrow G$ and $(\delta(C_2) \otimes \mu, R, \lceil \mathbf{q} \rceil)$ $\Longrightarrow^n_{NST} (G, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P$, then $(\eta; C_2, R, \lceil \mathbf{q} \rceil) \Longrightarrow^n_{NST} (G, Q)$.*

Proof. For all $R, G, I, P, Q, \mathbf{q}, C_2, n$ such that $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{closed}(Q)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C_2)$, $\mathbf{scl}(P)$, $\mathbf{Id} \Rightarrow R$ and $\mathbf{Id} \Rightarrow G$, we prove for all η, if $(\eta, R, \lceil \mathbf{q} \rceil) \Longrightarrow^n_{\text{NST}} (G, P)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta)$ and $(\delta(C_2) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longrightarrow^n_{\text{NST}} (G, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P$, then $(\eta; C_2, R, \lceil \mathbf{q} \rceil) \Longrightarrow^n_{\text{NST}} (G, Q)$ by induction on n.

- base case: $n = 0$.
- inductive case: $n = k + 1$.
  IH: for all η, if $(\eta, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, P)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$ and $(\delta(C_2) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P$, then $(\eta; C_2, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$.
  For all η such that $(\eta, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k+1}_{\text{NST}} (G, P)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta)$ and $(\delta(C_2) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k+1}_{\text{NST}} (G, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P$, by Lem. 211 we know $(\delta(C_2) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P$. To prove $(\eta; C_2, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k+1}_{\text{NST}} (G, Q)$, we need to prove
  - if $\eta; {C_2}^{(Stmt)}(\mathbf{skip}) > 0$, then $\eta; {C_2|_{\mathbf{skip}}}^{(State)} \models Q$.
    $\eta; {C_2}^{(Stmt)}(\mathbf{skip}) = \sum_\sigma \eta; C_2(\mathbf{skip}, \sigma) = 0$, which contradicts with $\eta; {C_2}^{(Stmt)}(\mathbf{skip}) > 0$.
  - $\eta; {C_2}^{(State)} \models \lceil \mathbf{q} \rceil$.
    From $(\eta, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k+1}_{\text{NST}} (G, P)$ we know $\eta^{(State)} \models \lceil \mathbf{q} \rceil$. By Lem. 201 we know $\eta; {C_2}^{(State)} = \eta^{(State)} \models \lceil \mathbf{q} \rceil$.

- for all η', if $\eta; C_2 \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta'$, then $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$.

  For all η' such that $\eta; C_2 \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta'$, by Lem. 209 there exists η'' such that $\eta' = \eta''; C_2$ and $\eta \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta''$. From $(\eta, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, P)$ we have $(\eta'', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, P)$. From $\eta \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta''$ by Lem. 188 we know $supp(\eta''^{(Stmt)}) \subseteq supp(\eta^{(Stmt)})$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta)$ by Lem. 290 we have $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta'')$. From $(\eta'', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, P)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta'')$ and $(\delta(C_2) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P$ by IH we have $(\eta''; C_2, R, \lceil \mathbf{q} \rceil)$
  $\Longmapsto^{k}_{\text{NST}} (G, Q)$, i.e., $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$.
- for all θ and η', if $\eta; C_2 \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, $\eta'^{(State)} \models \lceil \mathbf{q} \rceil$ and $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$.
  For all θ and η' such that $\eta; C_2 \hookrightarrow (\theta, \eta')$, from $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil)$ by Lem. 294 we know $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta; C_2)$. From $\eta; C_2 \hookrightarrow (\theta, \eta')$ and $\eta; C_2{}^{(State)} \models \lceil \mathbf{q} \rceil$ by Lem. 291 we have $\eta; C_2 \rightsquigarrow (\theta, \eta')$. There are three cases: $\eta^{(Stmt)}(\mathbf{skip}) = 1$, $\eta^{(Stmt)}(\mathbf{skip}) = 0$, or $0 < \eta^{(Stmt)}(\mathbf{skip}) < 1$.
  - $\eta^{(Stmt)}(\mathbf{skip}) = 1$.
    From $\eta^{(Stmt)}(\mathbf{skip}) = 1$ by Lem. 13 we know $\eta = \delta(\mathbf{skip}) \otimes \eta^{(State)}$. By Lem. 18 we have $\eta^{(Stmt)} = \delta(\mathbf{skip})$. From $(\eta; C_2) \rightsquigarrow (\theta, \eta')$ by Lem. 210 and Lem. 193 we know $\theta = \{(\sigma, \sigma) \mid \sigma \in supp(\eta^{(State)})\} \subseteq [\![\mathbf{Id}]\!] \subseteq [\![G]\!]$ and $\eta' = \delta(C_2) \otimes \eta^{(State)}$. From $(\eta, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, P)$ we know $\eta^{(State)} \models \lceil \mathbf{q} \rceil$. By Lem. 19 we know $\eta'^{(State)} = \eta^{(State)} \models \lceil \mathbf{q} \rceil$. From $\eta^{(Stmt)}(\mathbf{skip}) = 1$ by Lem. 199 we have $\eta = \eta|_{\mathbf{skip}}$, thus $\eta^{(State)} = \eta|_{\mathbf{skip}}{}^{(State)} \models P$. From $\eta^{(State)} \models \lceil \mathbf{q} \rceil$ we have $\eta^{(State)} \models \lceil \mathbf{q} \rceil \wedge P$. From $(\delta(C_2) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P$ we know $(\delta(C_2) \otimes \eta^{(State)}, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$, i.e., $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$.
  - From $(\eta; C_2) \rightsquigarrow (\theta, \eta')$ and $\eta^{(Stmt)}(\mathbf{skip}) = 0$ by Lem. 213 there exists η'' such that $\eta' = \eta''; C_2$ and $\eta \rightsquigarrow (\theta, \eta'')$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta)$ and $\eta^{(State)} \models \lceil \mathbf{q} \rceil$ by Lem. 291 we have $\eta \hookrightarrow (\theta, \eta'')$. From $(\eta, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, P)$ we know $\theta \subseteq [\![G]\!]$, $\eta''^{(State)} \models \lceil \mathbf{q} \rceil$ and $(\eta'', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, P)$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta)$ and $\eta \rightsquigarrow (\theta, \eta'')$ by Lem. 293 we know $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta'')$. From $(\eta'', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, P)$ and $(\delta(C_2) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P$ by IH we have $(\eta''_2; C_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$, i.e., $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$.
  - $0 < \eta^{(Stmt)}(\mathbf{skip}) < 1$.
    Let $p \overset{\text{def}}{=} \eta^{(Stmt)}(\mathbf{skip})$, then $0 < p < 1$. By Lem. 247 there exists η_1 and η_2 such that $\eta = \eta_1 \oplus_p \eta_2$, $\eta_1{}^{(Stmt)}(\mathbf{skip}) = 1$ and $\eta_2{}^{(Stmt)}(\mathbf{skip}) = 0$. By Lem. 248 we know $\eta; C_2 = (\eta_1 \oplus_p \eta_2); C_2 = (\eta_1; C_2) \oplus_p (\eta_2; C_2)$. From $\eta; C_2 \rightsquigarrow (\theta, \eta')$ we know $(\eta_1; C_2) \oplus_p (\eta_2; C_2) \rightsquigarrow (\theta, \eta')$. From $0 < p < 1$ by Lem. 245 there exists $\theta_1, \theta_2, \eta'_1, \eta'_2$ such

that $\theta = \theta_1 \cup \theta_2$, $\eta' = \eta'_1 \oplus_p \eta'_2$, $(\eta_1; C_2) \rightsquigarrow (\theta_1, \eta'_1)$, $(\eta_2; C_2) \rightsquigarrow (\theta_2, \eta'_2)$. From ${\eta_1}^{(Stmt)}(\mathbf{skip}) = 1$ by Lem. 13 we know $\eta_1 = \delta(\mathbf{skip}) \otimes {\eta_1}^{(State)}$. By Lem. 18 we have ${\eta_1}^{(Stmt)} = \delta(\mathbf{skip})$. From $(\eta_1; C_2) \rightsquigarrow (\theta_1, \eta'_1)$ by Lem. 210 and Lem. 193 we know $\theta_1 = \{(\sigma, \sigma) \mid \sigma \in supp({\eta_1}^{(State)})\}$ and $\eta'_1 = \delta(C_2) \otimes {\eta_1}^{(State)}$. From ${\eta_1}^{(Stmt)}(\mathbf{skip}) = 1$ by Lem. 192 we know $\eta_1 \rightsquigarrow (\{(\sigma, \sigma) \mid \sigma \in supp({\eta_1}^{(State)})\}, \delta(\mathbf{skip}) \otimes {\eta_1}^{(State)})$, i.e., $\eta_1 \rightsquigarrow (\theta_1, \eta_1)$. From $(\eta_2; C_2) \rightsquigarrow (\theta_2, \eta'_2)$ and ${\eta_2}^{(Stmt)}(\mathbf{skip}) = 0$ by Lem. 213 there exists η''_2 such that $\eta'_2 = \eta''_2; C_2$ and $\eta_2 \rightsquigarrow (\theta_2, \eta''_2)$, thus $\eta' = \eta'_1 \oplus_p \eta'_2 = (\delta(C_2) \otimes {\eta_1}^{(State)}) \oplus_p \eta''_2; C_2$. From $0 < p < 1$, $\eta_1 \rightsquigarrow (\theta_1, \eta_1)$ and $\eta_2 \rightsquigarrow (\theta_2, \eta''_2)$ by Lem. 246 we know $\eta_1 \oplus_p \eta_2 \rightsquigarrow (\theta_1 \cup \theta_2, \eta_1 \oplus_p \eta''_2)$, i.e., $\eta \rightsquigarrow (\theta, \eta_1 \oplus_p \eta''_2)$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta)$ and ${\eta}^{(State)} \models \lceil \mathbf{q} \rceil$ by Lem. 291 we have $\eta \hookrightarrow (\theta, \eta_1 \oplus_p \eta''_2)$. From $(\eta, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, P)$ we know $\theta \subseteq [\![G]\!]$, $(\eta_1 \oplus_p \eta''_2)^{(State)} \models \lceil \mathbf{q} \rceil$ and $(\eta_1 \oplus_p \eta''_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, P)$. By Lem. 12, Lem. 19 and Lem. 201 we know ${\eta'}^{(State)} = ((\delta(C_2) \otimes {\eta_1}^{(State)}) \oplus_p \eta''_2; C_2)^{(State)} = (\delta(C_2) \otimes {\eta_1}^{(State)})^{(State)} \oplus_p (\eta''_2; C_2)^{(State)} = {\eta_1}^{(State)} \oplus_p {\eta''_2}^{(State)} = (\eta_1 \oplus_p \eta''_2)^{(State)}$. From $(\eta_1 \oplus_p \eta''_2)^{(State)} \models \lceil \mathbf{q} \rceil$ we have ${\eta'}^{(State)} \models \lceil \mathbf{q} \rceil$. From $(\eta, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, P)$ and ${\eta}^{(Stmt)}(\mathbf{skip}) > 0$ we know ${\eta|_{\mathbf{skip}}}^{(State)} \models P$. From $0 < p < 1$ by Lem. 275 we know $supp(\eta) = supp(\eta_1 \oplus_p \eta_2) = supp(\eta_1) \cup supp(\eta_2) \supseteq supp(\eta_1)$. From ${\eta}^{(Stmt)}(\mathbf{skip}) > 0$ and ${\eta_1}^{(Stmt)}(\mathbf{skip}) = 1 > 0$ by Lem. 277 we know $supp(\eta_1|_{\mathbf{skip}}) \subseteq supp(\eta|_{\mathbf{skip}})$. By Lem. 24 we know $supp({\eta_1|_{\mathbf{skip}}}^{(State)}) \subseteq supp({\eta|_{\mathbf{skip}}}^{(State)})$. From ${\eta|_{\mathbf{skip}}}^{(State)} \models P$ and $\mathbf{scl}(P)$ we know ${\eta_1|_{\mathbf{skip}}}^{(State)} \models P$. From ${\eta_1}^{(Stmt)}(\mathbf{skip}) = 1$ by Lem. 199 we have $\eta_1 = \eta_1|_{\mathbf{skip}}$, thus ${\eta_1}^{(State)} = {\eta_1|_{\mathbf{skip}}}^{(State)} \models P$. From $(\eta, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, P)$ we know ${\eta}^{(State)} \models \lceil \mathbf{q} \rceil$. From $supp(\eta_1) \subseteq supp(\eta)$ by Lem. 24 we know $supp({\eta_1}^{(State)}) \subseteq supp({\eta}^{(State)})$. From ${\eta}^{(State)} \models \lceil \mathbf{q} \rceil$ and $\mathbf{scl}(\lceil \mathbf{q} \rceil)$ we know ${\eta_1}^{(State)} \models \lceil \mathbf{q} \rceil$. From ${\eta_1}^{(State)} \models P$ we have ${\eta_1}^{(State)} \models \lceil \mathbf{q} \rceil \wedge P$. From $(\delta(C_2) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P$ we know $(\delta(C_2) \otimes {\eta_1}^{(State)}, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$. From $supp(\eta_1) \subseteq supp(\eta)$ by Lem. 23 we know $supp({\eta_1}^{(Stmt)}) \subseteq supp({\eta}^{(Stmt)})$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta)$ by Lem. 290 we have $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$. Similarly we can prove $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_2)$. From $\eta_2 \rightsquigarrow (\theta_2, \eta''_2)$ by Lem. 293 we know $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta''_2)$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$ by Lem. 295 we know $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1 \oplus_p \eta''_2)$. From $0 < p < 1$, $(\eta_1 \oplus_p \eta''_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, P)$, $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{scl}(P)$ and $\mathbf{Id} \Rightarrow R$ by Lem. 297 we have $(\eta''_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta''_2)$ and $(\delta(C_2) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P$ by IH we have $(\eta''_2; C_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C_2)$ by Lem. 286 we know $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C_2) \otimes {\eta_1}^{(State)})$. From $0 < p < 1$, $(\delta(C_2) \otimes {\eta_1}^{(State)}, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$, $(\eta''_2; C_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$, $\mathbf{closed}(Q)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C_2)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C_2) \otimes {\eta_1}^{(State)})$ by Lem. 298 we know $((\delta(C_2) \otimes {\eta_1}^{(State)}) \oplus_p \eta''_2; C_2, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$, i.e., $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$.

Lemma 300. *For all* $R, G, \mathbf{q}, P, Q, b, C$, *if* $R, G, \lceil \mathbf{q} \rceil \models_{NST} \{P \wedge \lceil b \rceil\} C \{P\}$, $P \wedge \lceil \neg b \rceil \Rightarrow Q$, $\mathbf{closed}(Q)$, $\mathbf{Sta}(P, R, true)$, $\mathbf{Sta}(Q, R, true)$, $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C)$, $\mathbf{scl}(P)$, $\mathbf{Id} \Rightarrow R$ *and* $\mathbf{Id} \Rightarrow G$, *then* $R, G, \lceil \mathbf{q} \rceil \models_{NST} \{P\}$ ***while*** (b) ***do*** $C\{Q\}$.

Proof. For all $R, G, \mathbf{q}, P, Q, b, C$ such that $R, G, \lceil \mathbf{q} \rceil \models_{\text{NST}} \{P \wedge \lceil b \rceil\} C\{P\}$, $P \wedge \lceil \neg b \rceil \Rightarrow Q$, $\mathbf{closed}(Q)$, $\mathbf{Sta}(P, R, \text{true})$, $\mathbf{Sta}(Q, R, \text{true})$, $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C)$, $\mathbf{scl}(P)$, $\mathbf{Id} \Rightarrow R$, $\mathbf{Id} \Rightarrow G$, to prove $R, G, \lceil \mathbf{q} \rceil \models_{\text{NST}} \{P\}\mathbf{while}\ (b)\ \mathbf{do}\ C\{Q\}$, we need to prove for all μ and n, if $\mu \models \lceil \mathbf{q} \rceil \wedge P$, then $(\delta(\mathbf{while}\ (b)\ \mathbf{do}\ C, \mu) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{n}_{\text{NST}} (G, Q)$. For all μ and n such that $\mu \models \lceil \mathbf{q} \rceil \wedge P$, by Lem. 18 and Lem. 19 we know $\delta(\mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes \mu^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ and $\delta(\mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes \mu^{(State)} = \mu \models \lceil \mathbf{q} \rceil \wedge P$. To prove $(\delta(\mathbf{while}\ (b)\ \mathbf{do}\ C, \mu) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{n}_{\text{NST}} (G, Q)$, it suffices to prove for all η, if $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ and $\eta^{(State)} \models \lceil \mathbf{q} \rceil \wedge P$, then $(\eta, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{n}_{\text{NST}} (G, Q)$. We prove it by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all η, if $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ and $\eta^{(State)} \models \lceil \mathbf{q} \rceil \wedge P$, then $(\eta, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k}_{\text{NST}} (G, Q)$.
  For all η such that $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ and $\eta^{(State)} \models \lceil \mathbf{q} \rceil \wedge P$, to prove $(\eta, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k+1}_{\text{NST}} (G, Q)$, we need to prove
  - if $\eta^{(Stmt)}(\mathbf{skip}) > 0$, then $\eta|_{\mathbf{skip}}{}^{(State)} \models Q$.
    $\eta^{(Stmt)}(\mathbf{skip}) = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)(\mathbf{skip}) = 0$, which contradicts with $\eta^{(Stmt)}(\mathbf{skip}) > 0$.
  - $\eta^{(State)} \models \lceil \mathbf{q} \rceil$.
    From $\eta^{(State)} \models \lceil \mathbf{q} \rceil \wedge P$ we know $\eta^{(State)} \models \lceil \mathbf{q} \rceil$.
  - for all η', if $\eta \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta'$, then $(\eta', R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k}_{\square} (G, Q)$.
    For all η' such that $\eta \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta'$, by Lem. 188 we know $supp(\eta'^{(Stmt)}) \subseteq supp(\eta^{(Stmt)})$. From $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ by Lem. 27 we know $\eta'^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$. From $\mathbf{Sta}(P, R, \text{true})$ we know $\mathbf{Sta}(P, R, \lceil \mathbf{q} \rceil)$. From $\eta^{(State)} \models \lceil \mathbf{q} \rceil \wedge P$ and $\eta \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta'$ by Lem. 186 we have $\eta'^{(State)} \models \lceil \mathbf{q} \rceil \wedge P$. From $\eta'^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ by IH we have $(\eta', R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k}_{\text{NST}} (G, Q)$.
  - for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, $\eta'^{(State)} \models \lceil \mathbf{q} \rceil$ and $(\eta', R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k}_{\square} (G, Q)$.
    For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, from $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ by Lem. 190 we know $nextsplit(\eta) = \{nextsplit(\mathbf{while}\ (b)\ \mathbf{do}\ C)\} = \{\mathbf{split}(\text{true})\}$. From $\eta \hookrightarrow (\theta, \eta')$ by Lem. 191 we have $\eta \rightsquigarrow (\theta, \eta')$. There are three cases: $\eta^{(State)} \models \lceil b \rceil$, $\eta^{(State)} \models \lceil \neg b \rceil$ or $\eta^{(State)} \not\models \lceil b \rceil \wedge \eta^{(State)} \not\models \lceil \neg b \rceil$. We prove the three cases respectively.
    * $\eta^{(State)} \models \lceil b \rceil$.
      From $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$, $\eta^{(State)} \models \lceil b \rceil$ and $\eta \rightsquigarrow (\theta, \eta')$ by Lem. 219 and Lem. 193 we know $\theta = \{(\sigma, \sigma) \mid \sigma \in supp(\eta^{(State)})\} \subseteq$

$[\![\mathbf{Id}]\!] \subseteq [\![G]\!]$ and $\eta' = \delta(C; \mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes \eta^{(State)}$. By Lem. 217 we know $\eta' = \delta(C) \otimes \eta^{(State)}; \mathbf{while}\ (b)\ \mathbf{do}\ C$. By Lem. 19 we know $\eta'^{(State)} = \eta^{(State)} \models \lceil \mathbf{q} \rceil$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C)$ we know $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \mathbf{while}\ (b)\ \mathbf{do}\ C)$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \mathbf{while}\ (b)\ \mathbf{do}\ C)$ by Lem. 294 we know $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C) \otimes \eta^{(State)}; \mathbf{while}\ (b)\ \mathbf{do}\ C)$. From $R, G, \lceil \mathbf{q} \rceil \models_{\text{NST}} \{P \wedge \lceil b \rceil\} C \{P\}$ and $\eta^{(State)} \models \lceil \mathbf{q} \rceil \wedge P$ we know $(\delta(C) \otimes {\eta_1}^{(State)}, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, P)$. From IH we have $(\delta(\mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P$. From $(\delta(C) \otimes \eta^{(State)}, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, P)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C) \otimes \eta^{(State)})$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \mathbf{while}\ (b)\ \mathbf{do}\ C)$, $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{closed}(Q)$, $\mathbf{scl}(P)$, $\mathbf{Id} \Rightarrow R$ and $\mathbf{Id} \Rightarrow G$ by Lem. 299 we have $(\delta(C) \otimes \eta^{(State)}; \mathbf{while}\ (b)\ \mathbf{do}\ C, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$, i.e., $(\eta', R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$.

* $\eta^{(State)} \models \lceil \neg b \rceil$.
From $\eta^{(State)} \models \lceil \mathbf{q} \rceil \wedge P_2$ we know $\eta^{(State)} \models \lceil \mathbf{q} \rceil \wedge P_2 \wedge \lceil \neg b \rceil$. From $P_2 \wedge \lceil \neg b \rceil \Rightarrow Q$ we know $\eta^{(State)} \models \lceil \mathbf{q} \rceil \wedge Q$. From $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$, $\eta^{(State)} \models \lceil \neg b \rceil$ and $\eta \rightsquigarrow (\theta, \eta')$ by Lem. 220 and Lem. 193 we know $\theta = \{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\} \subseteq [\![\mathbf{Id}]\!] \subseteq [\![G]\!]$ and $\eta' = \delta(\mathbf{skip}) \otimes \eta^{(State)}$. By Lem. 18 and Lem. 19 we know $\eta'^{(Stmt)} = \delta(\mathbf{skip})$ and $\eta'^{(State)} = \eta^{(State)} \models \lceil \mathbf{q} \rceil \wedge Q$. From $\mathbf{Sta}(Q, R, \mathrm{true})$ and $I \Rightarrow \mathrm{true}$ we know $\mathbf{Sta}(Q, R, \lceil \mathbf{q} \rceil)$. From $\mathbf{Id} \Rightarrow G$, $\eta'^{(Stmt)} = \delta(\mathbf{skip})$, and $\eta'^{(State)} \models \lceil \mathbf{q} \rceil \wedge Q$ by Lem. 194 we know $(\eta', R, I) \Longrightarrow^k_{\text{NST}} (G_1 \vee G_2, Q)$.

* $\eta^{(State)} \not\models \lceil b \rceil \wedge \eta^{(State)} \not\models \lceil \neg b \rceil$.
From $\eta^{(State)} \not\models \lceil b \rceil$ by Lem. 226 we know $[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}} \neq 1$. From $\eta^{(State)} \not\models \lceil \neg b \rceil$ by Lem. 227 we know $[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}} \neq 0$, thus $0 < [\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}} < 1$. Let $p \stackrel{\text{def}}{=} [\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}$, then $0 < p < 1$. By Lem. 285 there exists η_1 and η_2 such that $\eta = \eta_1 \oplus_p \eta_2$, ${\eta_1}^{(State)} \models \lceil b \rceil$ and ${\eta_2}^{(State)} \models \lceil \neg b \rceil$. From $\eta \rightsquigarrow (\theta, \eta')$ we know $\eta_1 \oplus_p \eta_2 \rightsquigarrow (\theta, \eta')$. By Lem. 245 there exists $\theta_1, \theta_2, \eta'_1, \eta'_2$ such that $\theta = \theta_1 \cup \theta_2$, $\eta' = \eta'_1 \oplus_p \eta'$, $\eta_1 \rightsquigarrow (\theta_1, \eta'_1)$ and $\eta_2 \rightsquigarrow (\theta_2, \eta'_2)$. From $0 < p < 1$ by Lem. 275 we know $\mathit{supp}(\eta) = \mathit{supp}(\eta_1 \oplus_p \eta_2) = \mathit{supp}(\eta_1) \cup \mathit{supp}(\eta_2)$, thus $\mathit{supp}(\eta_1) \subseteq \mathit{supp}(\eta)$ and $\mathit{supp}(\eta_2) \subseteq \mathit{supp}(\eta)$. By Lem. 23 we know $\mathit{supp}({\eta_1}^{(Stmt)}) \subseteq \mathit{supp}(\eta^{(Stmt)})$ From $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ by Lem. 27 we know ${\eta_1}^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$. Similarly we can prove ${\eta_2}^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$. From ${\eta_1}^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$, ${\eta_1}^{(State)} \models \lceil b \rceil$ and $\eta_1 \rightsquigarrow (\theta_1, \eta'_1)$ by Lem. 219 and Lem. 193 we know $\theta_1 = \{(\sigma, \sigma) \mid \sigma \in \mathit{supp}({\eta_1}^{(State)})\}$ and $\eta'_1 = \delta(C; \mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes {\eta_1}^{(State)}$. From ${\eta_2}^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$, ${\eta_2}^{(State)} \models \lceil \neg b \rceil$ and $\eta_2 \rightsquigarrow (\theta_1, \eta'_1)$ by Lem. 220 and Lem. 193 we know $\theta_2 = \{(\sigma, \sigma) \mid \sigma \in \mathit{supp}({\eta_2}^{(State)})\}$ and $\eta'_2 = \delta(C; \mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes {\eta_2}^{(State)}$, thus $\theta = \theta_1 \cup \theta_2 \subseteq [\![\mathbf{Id}]\!] \subseteq [\![G]\!]$. By Lem. 217 we know $\eta' = \eta'_1 \oplus_p \eta'_2 = (\delta(C; \mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes {\eta_1}^{(State)}) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}) = (\delta(C) \otimes {\eta_1}^{(State)}; \mathbf{while}\ (b)\ \mathbf{do}\ C) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})$. By Lem. 12

and Lem. 19 we know $\eta'^{(State)} = {\eta_1}^{(State)} \oplus_p {\eta_2}^{(State)} = \eta^{(State)} \models \lceil \mathbf{q} \rceil$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C)$ we know $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \mathbf{while}\ (b)\ \mathbf{do}\ C)$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \mathbf{skip})$ by Lem. 286 we know $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C) \otimes {\eta_1}^{(State)})$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C) \otimes {\eta_1}^{(State)})$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \mathbf{while}\ (b)\ \mathbf{do}\ C)$ by Lem. 294 we know $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C) \otimes {\eta_1}^{(State)}; \mathbf{while}\ (b)\ \mathbf{do}\ C)$. From $\mathbf{scl}(\lceil \mathbf{q} \rceil)$ and $\mathbf{scl}(P)$ we know $\mathbf{scl}(\lceil \mathbf{q} \rceil \wedge P)$. From $supp(\eta_1) \subseteq supp(\eta)$ by Lem. 24 we know $supp({\eta_1}^{(State)}) \subseteq supp(\eta^{(State)})$. From $\eta^{(State)} \models \lceil \mathbf{q} \rceil \wedge P$ and $\mathbf{scl}(\lceil \mathbf{q} \rceil \wedge P)$ we know ${\eta_1}^{(State)} \models \lceil \mathbf{q} \rceil \wedge P$. From $R, G, \lceil \mathbf{q} \rceil \models_{\text{NST}} \{P \wedge \lceil b \rceil\} C \{P\}$ we know $(\delta(C) \otimes {\eta_1}^{(State)}, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, P)$. From IH we have $(\delta(\mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P$. From $(\delta(C) \otimes {\eta_1}^{(State)}, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, P)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C) \otimes {\eta_1}^{(State)})$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \mathbf{while}\ (b)\ \mathbf{do}\ C)$, $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{closed}(Q)$, $\mathbf{scl}(P)$, $\mathbf{Id} \Rightarrow R$ and $\mathbf{Id} \Rightarrow G$ by Lem. 299 we have $(\delta(C) \otimes {\eta_1}^{(State)}; \mathbf{while}\ (b)\ \mathbf{do}\ C, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$. From $supp(\eta_2) \subseteq supp(\eta)$ by Lem. 24 we know $supp({\eta_2}^{(State)}) \subseteq supp(\eta^{(State)})$. From $\eta^{(State)} \models \lceil \mathbf{q} \rceil \wedge P$ and $\mathbf{scl}(\lceil \mathbf{q} \rceil \wedge P)$ we know ${\eta_2}^{(State)} \models \lceil \mathbf{q} \rceil \wedge P$. From ${\eta_2}^{(State)} \models \lceil \neg b \rceil$ we know ${\eta_2}^{(State)} \models \lceil \mathbf{q} \rceil \wedge P \wedge \lceil \neg b \rceil$. From $P \wedge \lceil \neg b \rceil \Rightarrow Q$ we know ${\eta_2}^{(State)} \models \lceil \mathbf{q} \rceil \wedge Q$. From $\mathbf{Sta}(Q, R, \text{true})$ and $\lceil \mathbf{q} \rceil \Rightarrow \text{true}$ we know $\mathbf{Sta}(Q, R, \lceil \mathbf{q} \rceil)$. By Lem. 18 and Lem. 19 we know $(\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})^{(Stmt)} = \delta(\mathbf{skip})$ and $(\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})^{(State)} = {\eta_2}^{(State)} \models \lceil \mathbf{q} \rceil \wedge Q$. From $\mathbf{Sta}(Q, R, \lceil \mathbf{q} \rceil)$ and $\mathbf{Id} \Rightarrow G$ by Lem. 194 we know $(\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$. From $(\delta(C) \otimes {\eta_1}^{(State)}; \mathbf{while}\ (b)\ \mathbf{do}\ C, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$, $(\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q), 0 < p < 1, \mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C) \otimes {\eta_1}^{(State)}; \mathbf{while}\ (b)\ \mathbf{do}\ C)$ and $\mathbf{disablesplit}(\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)})$ and $\mathbf{closed}(Q)$ by Lem. 298 we have $((\delta(C) \otimes {\eta_1}^{(State)}; \mathbf{while}\ (b)\ \mathbf{do}\ C) \oplus_p (\delta(\mathbf{skip}) \otimes {\eta_2}^{(State)}), R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$, i.e., $(\eta', R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$.

Lemma 301. *For all $R, I, G, Q, \mathbf{q}, n, \eta$, if $\eta^{(State)} \models I$, $(\eta, R, \lceil \mathbf{q} \rceil) \Longrightarrow^n_{NST} (G, Q)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta)$, $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{Id} \Rightarrow G$, and $\forall x \in fv(I).\ G \Rightarrow \mathbf{Inv}(x)$, then $(\eta, R, I) \Longrightarrow^n_{NST} (G, Q)$.*

Proof. For all $R, I, G, Q, \mathbf{q}, n$ such that $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{Id} \Rightarrow G$ and $\forall x \in fv(I).\ G \Rightarrow \mathbf{Inv}(x)$, we prove for all η, if $\eta^{(State)} \models I$, $(\eta, R, \lceil \mathbf{q} \rceil) \Longrightarrow^n_{\text{NST}} (G, Q)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta)$, then $(\eta, R, I) \Longrightarrow^n_{\text{NST}} (G, Q)$ by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all η, if $\eta^{(State)} \models I$, $(\eta, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta)$, then $(\eta, R, I) \Longrightarrow^k_{\text{NST}} (G, Q)$.
  For all η such that $\eta^{(State)} \models I$, $(\eta, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k+1}_{\text{NST}} (G, Q)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$, to prove $(\eta, R, I) \Longrightarrow^{k+1}_{\text{NST}} (G, Q)$, we need to prove
  - if $\eta^{(Stmt)}(\mathbf{skip}) > 0$, then ${\eta|_{\mathbf{skip}}}^{(State)} \models Q$.
    From $(\eta, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k+1}_{\text{NST}} (G, Q)$ and $\eta^{(Stmt)}(\mathbf{skip}) > 0$ we know ${\eta|_{\mathbf{skip}}}^{(State)} \models Q$.

- $\eta^{(State)} \models I$.
  By assumption.
- for all η', if $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longmapsto^{k}_{\square} (G, Q)$.
  For all η' such that $\eta \overset{R}{\underset{I}{\rightarrowtail}} \eta'$, there exists η'' and b such that $\eta \overset{R}{\mapsto} \eta''$, $\eta''|_b = \eta'$ and $\eta'^{(State)} \models I$. From $\eta \overset{R}{\mapsto} \eta''$ and $\mathbf{sta}(\mathbf{q}, R)$ by Lem. 289 we have $\eta''^{(State)} \models \lceil \mathbf{q} \rceil$. From $\eta''|_b = \eta'$ by Lem. 20 we know $supp(\eta') \subseteq supp(\eta'')$. By Lem. 24 we know $supp(\eta'^{(State)}) \subseteq supp(\eta''^{(State)})$. From $\eta''^{(State)} \models \lceil \mathbf{q} \rceil$ and $\mathbf{scl}(\lceil \mathbf{q} \rceil)$ we know $\eta'^{(State)} \models \lceil \mathbf{q} \rceil$. From $\eta \overset{R}{\mapsto} \eta''$ and $\eta''|_b = \eta'$ we know $\eta \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta'$. From $(\eta, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ we know $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$. From $\eta \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta'$ by Lem. 188 we know $supp(\eta'^{(Stmt)}) \subseteq supp(\eta^{(Stmt)})$.
  From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta)$ by Lem. 290 we know $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta')$. From $\eta'^{(State)} \models I$, $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta')$ by IH we have $(\eta', R, I) \Longmapsto^{k}_{\text{NST}} (G, Q)$.
- for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, $\eta'^{(State)} \models I$ and $(\eta', R, I) \Longmapsto^{k}_{\square} (G, Q)$.
  For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, from $(\eta, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ we know $\eta^{(State)} \models \lceil \mathbf{q} \rceil$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta)$ and $\eta \hookrightarrow (\theta, \eta')$ by Lem. 291 we know $\eta \rightsquigarrow (\theta, \eta')$. From $(\eta, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ we know $\theta \subseteq [\![G]\!]$ and $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$. From $\theta \subseteq [\![G]\!]$ and $\forall x \in fv(I).\ G \Rightarrow \mathbf{Inv}(x)$ we know $\forall x \in fv(I), (\sigma, \sigma') \in \theta.\ \sigma'(x) = \sigma(x)$. From $\eta \rightsquigarrow (\theta, \eta')$ by Lem. 249 we know $\eta'^{(State)}|_{fv(I)} = \eta^{(State)}|_{fv(I)}$. From $\eta^{(State)} \models I$ by Lem. 272 we know $\eta'^{(State)} \models I$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta)$ and $\eta \rightsquigarrow (\theta, \eta')$ by Lem. 293 we have $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta')$. From $\eta'^{(State)} \models I$, $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta')$ by IH we have $(\eta', R, I) \Longmapsto^{k}_{\text{NST}} (G, Q)$.

Lemma 302. *For all $R, I, G, Q, \mathbf{q}, n, \eta, \eta_1, \mu, p$, if $0 < p < 1$, $\eta = \eta_1 \oplus_p (\delta(\textbf{\textit{skip}}) \otimes \mu)$, $\eta^{(State)} \models I$, $\mu \models Q$, $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{n}_{NST} (G, Q)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$, $\mathbf{closed}(Q)$, $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{Sta}(Q, R, true)$, $\mathbf{Id} \Rightarrow G$ and $\forall x \in fv(I).\ G \Rightarrow \mathbf{Inv}(x)$, then $(\eta, R, I) \Longmapsto^{n}_{NST} (G, Q)$.*

Proof. For all $R, I, G, Q, \mathbf{q}, n$ such that $\mathbf{closed}(Q)$, $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{Sta}(Q, R, \text{true})$, $\mathbf{Id} \Rightarrow G$ and $\forall x \in fv(I).\ G \Rightarrow \mathbf{Inv}(x)$, we prove for all η, η_1, μ, p, if $0 < p < 1$, $\eta = \eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \mu)$, $\eta^{(State)} \models I$, $\mu \models Q$, $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{n}_{\text{NST}} (G, Q)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$, then $(\eta, R, I) \Longmapsto^{n}_{\text{NST}} (G, Q)$ by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all η, η_1, μ, p, if $0 < p < 1$, $\eta = \eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \mu)$, $\eta^{(State)} \models I$, $\mu \models Q$, $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$, then $(\eta, R, I) \Longmapsto^{k}_{\text{NST}}$

(G, Q).
For all η, η_1, μ, p such that $0 < p < 1$, $\eta = \eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \mu)$, ${\eta}^{(State)} \models I$, $\mu \models Q$, $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$, to prove $(\eta, R, I) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$, we need to prove

- if ${\eta}^{(Stmt)}(\mathbf{skip}) > 0$, then ${\eta|_{\mathbf{skip}}}^{(State)} \models Q$.
  By Lem. 19 we know $(\delta(\mathbf{skip}) \otimes \mu)^{(Stmt)}(\mathbf{skip}) = \delta(\mathbf{skip})(\mathbf{skip}) = 1$. By Lem. 199 we know $(\delta(\mathbf{skip}) \otimes \mu)|_{\mathbf{skip}} = (\delta(\mathbf{skip}) \otimes \mu)$. Let $p_1 \stackrel{\text{def}}{=} {\eta_1}^{(Stmt)}(\mathbf{skip})$. There are two cases: $p_1 = 0$ or $p_1 > 0$. We prove the two cases respectively.
  * $p_1 = 0$.
    From $0 < p < 1$, ${\eta_1}^{(Stmt)}(\mathbf{skip}) = p_1 = 0$ and $(\delta(\mathbf{skip}) \otimes \mu)^{(Stmt)}(\mathbf{skip}) = 1 > 0$ by Lem. 288 we know $\eta|_{\mathbf{skip}} = (\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \mu))|_{\mathbf{skip}} = (\delta(\mathbf{skip}) \otimes \mu)|_{\mathbf{skip}} = (\delta(\mathbf{skip}) \otimes \mu)$. By Lem. 19 we know ${\eta|_{\mathbf{skip}}}^{(State)} = (\delta(\mathbf{skip}) \otimes \mu)^{(State)} = \mu$. From $\mu \models Q$ we know ${\eta|_{\mathbf{skip}}}^{(State)} \models Q$.
  * $p_1 > 0$.
    Let $p' \stackrel{\text{def}}{=} \frac{p \cdot p_1}{p \cdot p_1 + (1-p)}$. From $0 < p < 1$, ${\eta_1}^{(Stmt)}(\mathbf{skip}) = p_1 > 0$ and $(\delta(\mathbf{skip}) \otimes \mu)^{(Stmt)}(\mathbf{skip}) = 1 > 0$ by Lem. 288 we know $\eta|_{\mathbf{skip}} = (\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \mu))|_{\mathbf{skip}} = \eta_1|_{\mathbf{skip}} \oplus_{p'} (\delta(\mathbf{skip}) \otimes \mu)|_{\mathbf{skip}} = \eta_1|_{\mathbf{skip}} \oplus_{p'} (\delta(\mathbf{skip}) \otimes \mu)$. By Lem. 12 and Lem. 19 we know ${\eta|_{\mathbf{skip}}}^{(State)} = {\eta_1|_{\mathbf{skip}}}^{(State)} \oplus_{p'} (\delta(\mathbf{skip}) \otimes \mu)^{(State)} = {\eta_1|_{\mathbf{skip}}}^{(State)} \oplus_{p'} \mu$. From $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ and ${\eta_1}^{(Stmt)}(\mathbf{skip}) > 0$ we know ${\eta_1|_{\mathbf{skip}}}^{(State)} \models Q$. From $\mu \models Q$ and $\mathbf{closed}(Q)$ we know ${\eta_1|_{\mathbf{skip}}}^{(State)} \oplus_{p'} \mu \models Q$, i.e., ${\eta|_{\mathbf{skip}}}^{(State)} \models Q$.
- ${\eta}^{(State)} \models I$.
  By assumption.
- for all η', if $\eta \underset{I}{\overset{R}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longmapsto^{k}_{\square} (G, Q)$.
  For all η' such that $\eta \underset{I}{\overset{R}{\rightarrowtail}} \eta'$, i.e., $\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \mu) \underset{I}{\overset{R}{\rightarrowtail}} \eta'$. There exists η'' and b such that $\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \mu) \overset{R}{\mapsto} \eta''$, $\eta''|_b = \eta'$ and ${\eta'}^{(State)} \models I$.
  From $\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \mu) \overset{R}{\mapsto} \eta''$ and $0 < p < 1$ by Lem. 235 there exists η''_1, η''_2, p'' such that $0 < p'' < 1$, $\eta'' = \eta''_1 \oplus_{p''} \eta''_2$, $\eta_1 \overset{R}{\mapsto} \eta''_1$ and $\delta(\mathbf{skip}) \otimes \mu \overset{R}{\mapsto} \eta''_2$. From $\eta''|_b = \eta'$ by Lem. 205 we know $[\![\mathbf{Pr}(b)]\!]_{{\eta''}^{(State)}} > 0$. Let $p_1 \stackrel{\text{def}}{=} [\![\mathbf{Pr}(b)]\!]_{{\eta''_1}^{(State)}}$, $p_2 \stackrel{\text{def}}{=} [\![\mathbf{Pr}(b)]\!]_{{\eta''_2}^{(State)}}$. By Lem. 12 and Lem. 237 we know $[\![\mathbf{Pr}(b)]\!]_{{\eta''}^{(State)}} = [\![\mathbf{Pr}(b)]\!]_{(\eta''_1 \oplus_{p''} \eta''_2)^{(State)}} = [\![\mathbf{Pr}(b)]\!]_{{\eta''_1}^{(State)} \oplus_{p''} {\eta''_2}^{(State)}} = p'' \cdot [\![\mathbf{Pr}(b)]\!]_{{\eta''_1}^{(State)}} + (1-p'') \cdot [\![\mathbf{Pr}(b)]\!]_{{\eta''_2}^{(State)}} = p'' \cdot p_1 + (1-p'') \cdot p_2 > 0$.
  There are three cases: $p_1 > 0 \wedge p_2 = 0$, $p_1 = 0 \wedge p_2 > 0$ or $p_1 > 0 \wedge p_2 > 0$. We prove the three cases respectively.
  * $p_1 > 0 \wedge p_2 = 0$. From $0 < p < 1$, $[\![\mathbf{Pr}(b)]\!]_{{\eta''_1}^{(State)}} = p_1$ and $[\![\mathbf{Pr}(b)]\!]_{{\eta''_2}^{(State)}} = p_2$ by Lem. 238 we know $\eta' = \eta''|_b = (\eta''_1 \oplus_{p''} \eta''_2)|_b = \eta''_1|_b$. From $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ we know ${\eta_1}^{(State)} \models \lceil \mathbf{q} \rceil$.

From $\eta_1 \overset{R}{\mapsto} \eta_1''$ and $\mathbf{sta}(\mathbf{q}, R)$ by Lem. 289 we have ${\eta_1''}^{(State)} \models \lceil \mathbf{q} \rceil$. From $\eta_1''|_b = \eta'$ by Lem. 20 we know $supp(\eta') \subseteq supp(\eta_1'')$. By Lem. 24 we know $supp({\eta'}^{(State)}) \subseteq supp({\eta_1''}^{(State)})$. From ${\eta_1''}^{(State)} \models \lceil \mathbf{q} \rceil$ and $\mathbf{scl}(\lceil \mathbf{q} \rceil)$ we know ${\eta_1''}^{(State)} \models \lceil \mathbf{q} \rceil$. From $\eta_1 \overset{R}{\mapsto} \eta_1''$ and $\eta_1''|_b = \eta'$ we know $\eta_1 \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta'$. From $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ we know $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$. From $\eta_1 \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta'$ by Lem. 188 we know $supp({\eta'}^{(Stmt)}) \subseteq supp({\eta_1}^{(Stmt)})$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$ by Lem. 290 we know $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta')$. From ${\eta'}^{(State)} \models I$, $(\eta', R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta')$, $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{Id} \Rightarrow G$ and $\forall x \in fv(I).\ G \Rightarrow \mathbf{Inv}(x)$ by Lem. 301 we have $(\eta', R, I) \Longmapsto^{k}_{\text{NST}} (G, Q)$.

* $p_1 = 0 \wedge p_2 > 0$.
From $0 < p < 1$, $[\![\mathbf{Pr}(b)]\!]_{{\eta_1''}^{(State)}} = p_1$ and $[\![\mathbf{Pr}(b)]\!]_{{\eta_2''}^{(State)}} = p_2$ by Lem. 238 we know $\eta' = \eta''|_b = (\eta_1'' \oplus_{p''} \eta_2'')|_b = \eta_2''|_b$. From $\delta(\mathbf{skip}) \otimes \mu \overset{R}{\mapsto} \eta_2''$ and $\eta_2''|_b = \eta'$ we know $\delta(\mathbf{skip}) \otimes \mu \overset{R}{\underset{\text{true}}{\rightarrowtail}} \eta'$. By Lem. 188 we know $supp({\eta'}^{(Stmt)}) \subseteq supp((\delta(\mathbf{skip}) \otimes \mu)^{(Stmt)})$. By Lem. 18 we know $(\delta(\mathbf{skip}) \otimes \mu)^{(Stmt)} = \delta(\mathbf{skip})$. By Lem. 27 we know ${\eta'}^{(Stmt)} = \delta(\mathbf{skip})$. From $\delta(\mathbf{skip}) \otimes \mu \overset{R}{\underset{\text{true}}{\rightarrowtail}} \eta'$, $\mu \models Q$ and $\mathbf{Sta}(Q, R, \text{true})$ by Lem. 186 we have ${\eta'}^{(State)} \models Q$. From ${\eta'}^{(State)} \models I$ we know ${\eta'}^{(State)} \models I \wedge Q$. From $\mathbf{Sta}(Q, R, \text{true})$ and $I \Rightarrow \text{true}$ we have $\mathbf{Sta}(Q, R, I)$. From $\mathbf{Id} \Rightarrow G$, ${\eta'}^{(Stmt)} = \delta(\mathbf{skip})$, ${\eta'}^{(State)} \models I \wedge Q$ by Lem. 194 we know $(\eta', R, I) \Longmapsto^{k}_{\text{NST}} (G, Q)$.

* $p_1 > 0 \wedge p_2 > 0$.
Let $p' \overset{\text{def}}{=} \frac{p'' \cdot p_1}{p'' \cdot p_1 + (1 - p'') \cdot p_2}$. From $0 < p'' < 1$, $p_1 > 0$ and $p_2 > 0$ we know $0 < p' < 1$. From $0 < p < 1$, $[\![\mathbf{Pr}(b)]\!]_{{\eta_1''}^{(State)}} = p_1$ and $[\![\mathbf{Pr}(b)]\!]_{{\eta_2''}^{(State)}} = p_2$ by Lem. 238 we know $\eta' = \eta''|_b = (\eta_1'' \oplus_{p''} \eta_2'')|_b = \eta_1''|_b \oplus_{p'} \eta_2''|_b$. From $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ we know ${\eta_1}^{(State)} \models \lceil \mathbf{q} \rceil$. From $\eta_1 \overset{R}{\mapsto} \eta_1''$ and $\mathbf{sta}(\mathbf{q}, R)$ by Lem. 289 we have ${\eta_1''}^{(State)} \models \lceil \mathbf{q} \rceil$. By Lem. 20 we know $supp(\eta_1''|_b) \subseteq supp(\eta_1'')$. By Lem. 24 we know $supp({\eta_1''|_b}^{(State)}) \subseteq supp({\eta_1''}^{(State)})$. From ${\eta_1''}^{(State)} \models \lceil \mathbf{q} \rceil$ and $\mathbf{scl}(\lceil \mathbf{q} \rceil)$ we know ${\eta_1''|_b}^{(State)} \models \lceil \mathbf{q} \rceil$. From $\eta_1 \overset{R}{\mapsto} \eta_1''$ we know $\eta_1 \overset{R}{\underset{\lceil \mathbf{q} \rceil}{\rightarrowtail}} \eta_1''|_b$. From $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k+1}_{\text{NST}} (G, Q)$ we know $(\eta_1''|_b, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G, Q)$. From $\delta(\mathbf{skip}) \otimes \mu \overset{R}{\mapsto} \eta_2''$ by Lem. 188 we know $supp({\eta_2''}^{(Stmt)}) \subseteq supp((\delta(\mathbf{skip}) \otimes \mu)^{(Stmt)})$. By Lem. 18 we know $(\delta(\mathbf{skip}) \otimes \mu)^{(Stmt)} = \delta(\mathbf{skip})$. By Lem. 27 we know ${\eta_2''}^{(Stmt)} = \delta(\mathbf{skip})$. By Lem. 13 we know $\eta_2'' = \delta(\mathbf{skip}) \otimes {\eta_2''}^{(State)}$. From $[\![\mathbf{Pr}(b)]\!]_{{\eta_2''}^{(State)}} = p_2 > 0$ by

Lem. 242 and Lem. 206 we know $\eta_2''|_b = (\delta(\mathbf{skip}) \otimes \eta_2''^{(State)})|_b = \delta(\mathbf{skip}) \otimes \eta_2''^{(State)}|_b = \delta(\mathbf{skip}) \otimes \eta_2''|_b{}^{(State)}$, thus $\eta' = \eta_1''|_b \oplus_{p'} \eta_2''|_b = \eta_1''|_b \oplus_{p'} (\delta(\mathbf{skip}) \otimes \eta_2''|_b{}^{(State)})$. From $\delta(\mathbf{skip}) \otimes \mu \overset{R}{\mapsto} \eta_2''$ we know $\delta(\mathbf{skip}) \otimes \mu \underset{\text{true}}{\overset{R}{\rightarrowtail}} \eta_2''|_b$. From $\mu \models Q$ and $\mathbf{Sta}(Q, R, \text{true})$ by Lem. 186 we have $\eta_2''|_b{}^{(State)} \models Q$. From $\eta_1 \underset{\lceil \mathbf{q} \rceil}{\overset{R}{\rightarrowtail}} \eta_1''|_b$ by Lem. 188 we know $supp(\eta_1''|_b{}^{(Stmt)}) \subseteq supp(\eta_1{}^{(Stmt)})$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$ by Lem. 290 we know $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1''|_b)$. From $0 < p' < 1$, $\eta' = \eta_1''|_b \oplus_{p'} (\delta(\mathbf{skip}) \otimes \eta_2''|_b{}^{(State)})$, $\eta'^{(State)} \models I, \eta_2''|_b{}^{(State)} \models Q, (\eta_1''|_b, R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1''|_b)$ by IH we have $(\eta', R, I) \Longrightarrow^k_{\text{NST}} (G, Q)$.

- for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!], \eta'^{(State)} \models I$ and $(\eta', R, I) \Longrightarrow^k_{\square} (G, Q)$.
  For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, from $0 < p < 1$ by Lem. 243 and Lem. 190 we know $nextsplit(\eta) = nextsplit(\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \mu)) = nextsplit(\eta_1) \cup nextsplit(\delta(\mathbf{skip}) \otimes \mu) \supseteq nextsplit(\delta(\mathbf{skip}) \otimes \mu) = \{nextsplit(\mathbf{skip})\} = \{\mathbf{split}(\text{true})\}$. From $\eta \hookrightarrow (\theta, \eta')$ by Lem. 191 we know $\eta \rightsquigarrow (\theta, \eta')$, i.e., $\eta_1 \oplus_p (\delta(\mathbf{skip}) \otimes \mu) \rightsquigarrow (\theta, \eta')$. From $0 < p < 1$ by Lem. 245 there exists $\theta_1, \theta_2, \eta_1', \eta_2'$ such that $\theta = \theta_1 \cup \theta_2$, $\eta' = \eta_1' \oplus_p \eta_2'$, $\eta_1 \rightsquigarrow (\theta_1, \eta_1')$, $\delta(\mathbf{skip}) \otimes \mu \rightsquigarrow (\theta_2, \eta_2')$. From $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k+1}_{\text{NST}} (G, Q)$ we know $\eta_1{}^{(State)} \models \lceil \mathbf{q} \rceil$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$ and $\eta_1 \rightsquigarrow (\theta_1, \eta_1')$ by Lem. 291 we know $\eta_1 \hookrightarrow (\theta_1, \eta_1')$. From $(\eta_1, R, \lceil \mathbf{q} \rceil) \Longrightarrow^{k+1}_{\text{NST}} (G, Q)$ we know $\theta_1 \subseteq [\![G]\!]$ and $(\eta_1', R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$. By Lem. 18 and Lem. 19 we have $\delta(\mathbf{skip}) \otimes \mu^{(Stmt)} = \delta(\mathbf{skip})$ and $\delta(\mathbf{skip}) \otimes \mu^{(State)} = \mu$. From $\delta(\mathbf{skip}) \otimes \mu \rightsquigarrow (\theta_2, \eta_2')$ by Lem. 192 and Lem. 193 we know $\theta_2 = \{(\sigma, \sigma) \mid \sigma \in \mu\} \subseteq [\![\mathbf{Id}]\!] \subseteq [\![G]\!]$ and $\eta_2' = \delta(\mathbf{skip}) \otimes \mu$, thus $\theta = \theta_1 \cup \theta_2 \subseteq [\![G]\!]$ and $\eta' = \eta_1' \oplus_p \eta_2' = \eta_1' \oplus_p (\delta(\mathbf{skip}) \otimes \mu)$. From $\theta \subseteq [\![G]\!]$ and $\forall x \in fv(I).\ G \Rightarrow \mathbf{Inv}(x)$ we know $\forall x \in fv(I), (\sigma, \sigma') \in \theta.\ \sigma'(x) = \sigma(x)$. From $\eta \rightsquigarrow (\theta, \eta')$ by Lem. 249 we know $\eta'^{(State)}|_{fv(I)} = \eta^{(State)}|_{fv(I)}$. From $\eta^{(State)} \models I$ by Lem. 272 we know $\eta'^{(State)} \models I$. From $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1)$ and $\eta_1 \rightsquigarrow (\theta_1, \eta_1')$ by Lem. 293 we have $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1')$. From $0 < p < 1$, $\eta' = \eta_1' \oplus_p (\delta(\mathbf{skip}) \otimes \mu)$, $\eta'^{(State)} \models I$, $\mu \models Q$, $(\eta_1', R, \lceil \mathbf{q} \rceil) \Longrightarrow^k_{\text{NST}} (G, Q)$ and $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta_1')$ by IH we have $(\eta', R, I) \Longrightarrow^k_{\text{NST}} (G, Q)$.

Lemma 303 (Soundness of (WHILE-NST) rule). *For all* $b, C, R, G_1, G_2, I, P_1, P_2, Q, \mathbf{q}$, *if* $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $\mathbf{Sta}(P_2, R, true)$, $P_1 \Rightarrow \lceil b \rceil$, $\mathbf{Sta}(Q, R, true)$, $P_2 \wedge \lceil \neg b \rceil \Rightarrow Q$, $R, G_1, I \models_{ST} \{P_1\} C \{P_1 \vee P_2\}$, $P_2 \wedge \lceil b \rceil \Rightarrow \lceil \mathbf{q} \rceil$, $R, G_2, \lceil \mathbf{q} \rceil \models_{NST} \{P_2 \wedge \lceil b \rceil\} C \{P_2\}$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C)$, $\mathbf{sta}(q, R)$, $\mathbf{closed}(Q)$, $\mathbf{scl}(P_2)$, $\mathbf{Id} \Rightarrow R$, $\mathbf{Id} \Rightarrow G_1$, $\mathbf{Id} \Rightarrow G_2$ *and* $\forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x)$, *then* $R, G_1 \vee G_2, I \models_{NST} \{P_1 \vee P_2\}$ ***while*** (b) ***do*** $C\{Q\}$.

Proof. For all $b, C, R, G_1, G_2, I, P_1, P_2, Q, \mathbf{q}$, such that $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $\mathbf{Sta}(P_2, R, \text{true})$, $P_1 \Rightarrow \lceil b \rceil$, $\mathbf{Sta}(Q, R, \text{true})$, $P_2 \wedge \lceil \neg b \rceil \Rightarrow Q$, $R, G_1, I \models_{\text{ST}} \{P_1\} C \{P_1 \vee P_2\}$, $P_2 \wedge \lceil b \rceil \Rightarrow \lceil \mathbf{q} \rceil$, $R, G_2, \lceil \mathbf{q} \rceil \models_{\text{NST}} \{P_2 \wedge \lceil b \rceil\} C \{P_2\}$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C)$, $\mathbf{sta}(q, R)$,

$\mathbf{closed}(Q)$ and $\forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x)$, to prove $R, G_1 \vee G_2, I \models_{\text{NST}} \{P_1 \vee P_2\}\mathbf{while}\ (b)\ \mathbf{do}\ C\{Q\}$, we need to prove for all n and μ, if $\mu \models I \wedge (P_1 \vee P_2)$, then $(\mathit{init}(\mathbf{while}\ (b)\ \mathbf{do}\ C, \mu), R, I) \Longmapsto^n_{\text{NST}} (G_1 \vee G_2, Q)$. For all n and μ such that $\mu \models I \wedge (P_1 \vee P_2)$, by Lem. 18 we know $\mathit{init}(\mathbf{while}\ (b)\ \mathbf{do}\ C, \mu)^{(Stmt)} = (\delta(\mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes \mu)^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$. To prove $(\mathit{init}(\mathbf{while}\ (b)\ \mathbf{do}\ C, \mu), R, I) \Longmapsto^n_{\text{NST}} (G_1 \vee G_2, Q)$, it suffices to prove for all η, if $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$, then $(\eta, R, I) \Longmapsto^n_{\text{NST}} (G_1 \vee G_2, Q)$. We prove it by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all η, if $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ and $\eta^{(State)} \models I \wedge (P_1 \vee P_2)$, then $(\eta, R, I) \Longmapsto^k_{\text{NST}} (G_1 \vee G_2, Q)$.
  For all η such that $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ and $\eta^{(State)} \models I \wedge (P_1 \vee P_2)$, to prove $(\eta, R, I) \Longmapsto^{k+1}_{\text{NST}} (G_1 \vee G_2, Q)$, we need to prove
  - if $\eta^{(Stmt)}(\mathbf{skip}) > 0$, then $\eta|_{\mathbf{skip}}{}^{(State)} \models Q$.
    $\eta^{(Stmt)}(\mathbf{skip}) = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)(\mathbf{skip}) = 0$, which contradicts with $\eta^{(Stmt)}(\mathbf{skip}) > 0$.
  - $\eta^{(State)} \models I$.
    From $\eta^{(State)} \models I \wedge (P_1 \vee P_2)$ we know $\eta^{(State)} \models I$.
  - for all η', if $\eta \underset{I}{\overset{R}{\rightarrowtail}} \eta'$, then $(\eta', R, I) \Longmapsto^k_{\text{NST}} (G_1 \vee G_2, Q)$.
    For all η' such that $\eta \underset{I}{\overset{R}{\rightarrowtail}} \eta'$, by Lem. 188 we know $\mathit{supp}(\eta'^{(Stmt)}) \subseteq \mathit{supp}(\eta^{(Stmt)})$. From $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ by Lem. 27 we know $\eta'^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$. From $\eta^{(State)} \models I$, $\eta \underset{I}{\overset{R}{\rightarrowtail}} \eta'$ and $\mathbf{Sta}(P_1 \vee P_2, R, I)$ by Lem. 186 we know $\eta'^{(State)} \models I \wedge (P_1 \vee P_2)$. By IH we have $(\eta', R, I) \Longmapsto^k_{\text{NST}} (G_1 \vee G_2, Q)$.
  - for all θ and η', if $\eta \hookrightarrow (\theta, \eta')$, then $\theta \subseteq [\![G]\!]$, $\eta'^{(State)} \models I$ and $(\eta', R, I) \Longmapsto^k_{\text{NST}} (G_1 \vee G_2, Q)$.
    For all θ and η' such that $\eta \hookrightarrow (\theta, \eta')$, from $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ by Lem. 190 we know $\mathit{nextsplit}(\eta) = \{\mathit{nextsplit}(\mathbf{while}\ (b)\ \mathbf{do}\ C)\} = \{\mathbf{split}(\mathrm{true})\}$. From $\eta \hookrightarrow (\theta, \eta')$ by Lem. 191 we have $\eta \rightsquigarrow (\theta, \eta')$. From $\eta^{(State)} \models I \wedge (P_1 \vee P_2)$ we know $\eta^{(State)} \models I \wedge P_1$ or $\eta^{(State)} \models I \wedge P_2$. We prove the two cases respectively.
    * $\eta^{(State)} \models I \wedge P_1$.
      From $P_1 \Rightarrow \lceil b \rceil$ we know $\eta^{(State)} \models \lceil b \rceil$. From $\eta^{(Stmt)} = \delta(\mathbf{while}\ (b)\ \mathbf{do}\ C)$ by Lem. 219 and Lem. 193 we know $\theta = \{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\} \subseteq [\![\mathbf{Id}]\!] \subseteq [\![G_1 \vee G_2]\!]$ and $\eta' = \delta(C; \mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes \eta^{(State)}$. By Lem. 217 we know $\eta' = (\delta(C) \otimes \eta^{(State)}); \mathbf{while}\ (b)\ \mathbf{do}\ C$. From $R, G_1, I \models_{\text{ST}} \{P_1\}C\{P_1 \vee P_2\}$ and $\eta^{(State)} \models I \wedge P_1$ we know $(\delta(C) \otimes \eta^{(State)}, R, I) \Longmapsto^k_{\text{ST}} (G_1, P_1 \vee P_2)$. By IH we know $(\delta(\mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes \mu, R, I) \Longmapsto^k_{\text{ST}} (G_1 \vee G_2, Q)$ for all μ such that $\mu \models I \wedge (P_1 \vee P_2)$. From $\mathbf{Id} \Rightarrow G_1 \vee G_2$ and $(\delta(C) \otimes \eta^{(State)}, R, I) \Longmapsto^k_{\text{ST}} (G_1, P_1 \vee P_2)$ by Lem. 216 we know $((\delta(C) \otimes \eta^{(State)}); \mathbf{while}\ (b)\ \mathbf{do}\ C, R, I) \Longmapsto^k_{\text{ST}} (G_1 \vee G_2, Q)$, i.e., $(\eta', R, I) \Longmapsto^k_{\text{ST}} (G_1 \vee G_2, Q)$. By Lem. 179 we have $(\eta', R, I) \Longmapsto^k_{\text{NST}} (G_1 \vee G_2, Q)$.

* $\eta^{(State)} \models I \wedge P_2$.
  There are three cases: $\eta^{(State)} \models \lceil b \rceil$, $\eta^{(State)} \models \lceil \neg b \rceil$ or $\eta^{(State)} \not\models \lceil b \rceil \wedge \eta^{(State)} \not\models \lceil \neg b \rceil$. We prove the three cases respectively.
  - $\eta^{(State)} \models \lceil b \rceil$.
    From $\eta^{(Stmt)} = \delta(\textbf{while}\ (b)\ \textbf{do}\ C)$, $\eta^{(State)} \models \lceil b \rceil$ and $\eta \rightsquigarrow (\theta, \eta')$ by Lem. 219 and Lem. 193 we know $\theta = \{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\} \subseteq [\![\textbf{Id}]\!] \subseteq [\![G_1 \vee G_2]\!]$ and $\eta' = \delta(C; \textbf{while}\ (b)\ \textbf{do}\ C) \otimes \eta^{(State)}$. By Lem. 19 we know $\eta'^{(State)} = \eta^{(State)} \models I$. To prove $(\eta', R, I) \Longmapsto^k_{\text{NST}} (G_1 \vee G_2, Q)$, from $G_2 \Rightarrow G_1 \vee G_2$ by Lem. 181, it suffices to prove $(\eta', R, I) \Longmapsto^k_{\text{NST}} (G_2, Q)$. From $\textbf{disablesplit}(\lceil \mathbf{q} \rceil, C)$ we know $\textbf{disablesplit}(\lceil \mathbf{q} \rceil,$ $\textbf{while}\ (b)\ \textbf{do}\ C)$, thus $\textbf{disablesplit}(\lceil \mathbf{q} \rceil, C; \textbf{while}\ (b)\ \textbf{do}\ C)$. By Lem. 286 we know $\textbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C; \textbf{while}\ (b)\ \textbf{do}\ C) \otimes \eta^{(State)})$, i.e., $\textbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta')$. To prove $(\eta', R, I) \Longmapsto^k_{\text{NST}} (G_2, Q)$, from $\eta'^{(State)} \models I$, $\textbf{disablesplit}(\lceil \mathbf{q} \rceil, \eta')$, $\textbf{sta}(\mathbf{q}, R)$, $\textbf{Id} \Rightarrow G_2$ and $\forall x \in fv(I).\ G_2 \Rightarrow \textbf{Inv}(x)$ by Lem. 301, it suffices to prove $(\eta', R, \lceil \mathbf{q} \rceil), \Longmapsto^k_{\text{NST}} (G_2, Q)$, i.e., $(\delta(C; \textbf{while}\ (b)\ \textbf{do}\ C) \otimes \eta^{(State)}, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G_2, Q)$. By Lem. 217 we know $\delta(C; \textbf{while}\ (b)\ \textbf{do}\ C) \otimes \eta^{(State)} = \delta(C) \otimes \eta^{(State)}; \textbf{while}\ (b)\ \textbf{do}\ C$, thus we need to prove $(\delta(C) \otimes \eta^{(State)}; \textbf{while}\ (b)\ \textbf{do}\ C), R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G_2, Q)$. From $\textbf{disablesplit}(\lceil \mathbf{q} \rceil, C)$ by Lem. 286 we know $\textbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C) \otimes \eta^{(State)})$. From $\eta^{(State)} \models P_2$ and $\eta^{(State)} \models \lceil b \rceil$ we know $\eta^{(State)} \models P_2 \wedge \lceil b \rceil$. From $P_2 \wedge \lceil b \rceil \Rightarrow \lceil bfq \rceil$ we know $\eta^{(State)} \models \lceil \mathbf{q} \rceil \wedge P_2 \wedge \lceil b \rceil$. From $R, G_2, \lceil \mathbf{q} \rceil \models_{\text{NST}} \{P_2 \wedge \lceil b \rceil\} C \{P_2\}$ we know $(\delta(C) \otimes \eta^{(State)}, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G_2, P_2)$. To prove $(\delta(C) \otimes \eta^{(State)}; \textbf{while}\ (b)\ \textbf{do}\ C, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G_2, Q)$, from $(\delta(C) \otimes \eta^{(State)}, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G_2, P_2)$, $\textbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C) \otimes \eta^{(State)})$, $\textbf{disablesplit}(\textbf{while}\ (b)\ \textbf{do}\ C)$, $\textbf{sta}(\mathbf{q}, R)$, $\textbf{closed}(Q)$, $\textbf{scl}(P_2)$, $\textbf{Id} \Rightarrow R$ and $\textbf{Id} \Rightarrow G_2$ by Lem. 299, it suffices to prove $(\delta(\textbf{while}\ (b)\ \textbf{do}\ C) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G_2, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P_2$. From $R, G_2, \lceil \mathbf{q} \rceil \models_{\text{NST}} \{P_2 \wedge \lceil b \rceil\} C \{P_2\}$, $P_2 \wedge \lceil \neg b \rceil \Rightarrow Q$, $\textbf{closed}(Q)$, $\textbf{Sta}(P_2, R, \text{true})$, $\textbf{Sta}(Q, R, \text{true})$, $\textbf{sta}(\mathbf{q}, R)$, $\textbf{disablesplit}(\lceil \mathbf{q} \rceil, C)$, $\textbf{scl}(P_2)$, $\textbf{Id} \Rightarrow R$ and $\textbf{Id} \Rightarrow G_2$ by Lem. 300 we know $R, G_2, \lceil \mathbf{q} \rceil \models_{\text{NST}} \{P_2\} \textbf{while}\ (b)\ \textbf{do}\ C \{Q\}$, thus $(\delta(\textbf{while}\ (b)\ \textbf{do}\ C) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G_2, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P_2$.
  - $\eta^{(State)} \models \lceil \neg b \rceil$.
    From $\eta^{(State)} \models I \wedge P_2$ we know $\eta^{(State)} \models I \wedge P_2 \wedge \lceil \neg b \rceil$. From $P_2 \wedge \lceil \neg b \rceil \Rightarrow Q$ we know $\eta^{(State)} \models I \wedge Q$. From $\eta^{(Stmt)} = \delta(\textbf{while}\ (b)\ \textbf{do}\ C)$, $\eta^{(State)} \models \lceil \neg b \rceil$ and $\eta \rightsquigarrow (\theta, \eta')$ by Lem. 220 and Lem. 193 we know $\theta = \{(\sigma, \sigma) \mid \sigma \in \mathit{supp}(\eta^{(State)})\} \subseteq [\![\textbf{Id}]\!] \subseteq [\![G_1 \vee G_2]\!]$ and $\eta' = \delta(\textbf{skip}) \otimes \eta^{(State)}$. By Lem. 18 and Lem. 19 we know $\eta'^{(Stmt)} = \delta(\textbf{skip})$ and $\eta'^{(State)} = \eta^{(State)} \models I \wedge Q$. From $\textbf{Sta}(Q, R, \text{true})$ and $I \Rightarrow \text{true}$ we know $\textbf{Sta}(Q, R, I)$. From $\textbf{Id} \Rightarrow G_1 \vee G_2$, $\eta'^{(Stmt)} = \delta(\textbf{skip})$, and $\eta'^{(State)} = \eta^{(State)} \models I \wedge Q$ by Lem. 194 we know $(\eta', R, I) \Longmapsto^k_{\text{NST}} (G_1 \vee G_2, Q)$.

· $\eta^{(State)} \not\models \lceil b \rceil \wedge \eta^{(State)} \not\models \lceil \neg b \rceil$.
From $\eta^{(State)} \not\models \lceil b \rceil$ by Lem. 226 we know $[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}} \neq 1$. From $\eta^{(State)} \not\models \lceil \neg b \rceil$ by Lem. 227 we know $[\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}} \neq 0$, thus $0 < [\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}} < 1$. Let $p \stackrel{\text{def}}{=} [\![\mathbf{Pr}(b)]\!]_{\eta^{(State)}}$, then $0 < p < 1$. By Lem. 285 there exists η_1 and η_2 such that $\eta = \eta_1 \oplus_p \eta_2$, ${\eta_1}^{(State)} \models \lceil b \rceil$ and ${\eta_2}^{(State)} \models \lceil \neg b \rceil$. From $\eta \rightsquigarrow (\theta, \eta')$ we know $\eta_1 \oplus_p \eta_2 \rightsquigarrow (\theta, \eta')$. By Lem. 245 there exists $\theta_1, \theta_2, \eta_1', \eta_2'$ such that $\theta = \theta_1 \cup \theta_2$, $\eta' = \eta_1' \oplus_p \eta'$, $\eta_1 \rightsquigarrow (\theta_1, \eta_1')$ and $\eta_2 \rightsquigarrow (\theta_2, \eta_2')$. From $0 < p < 1$ by Lem. 275 we know $supp(\eta) = supp(\eta_1 \oplus_p \eta_2) = supp(\eta_1) \cup supp(\eta_2) \supseteq supp(\eta_1)$. By Lem. 23 we know $supp({\eta_1}^{(Stmt)}) \subseteq supp(\eta^{(Stmt)})$ From $\eta^{(Stmt)} = \delta(\textbf{while}\ (b)\ \textbf{do}\ C)$ by Lem. 27 we know ${\eta_1}^{(Stmt)} = \delta(\textbf{while}\ (b)\ \textbf{do}\ C)$. Similarly we can prove ${\eta_2}^{(Stmt)} = \delta(\textbf{while}\ (b)\ \textbf{do}\ C)$. From ${\eta_1}^{(Stmt)} = \delta(\textbf{while}\ (b)\ \textbf{do}\ C)$, ${\eta_1}^{(State)} \models \lceil b \rceil$ and $\eta_1 \rightsquigarrow (\theta_1, \eta_1')$ by Lem. 219 and Lem. 193 we know $\theta_1 = \{(\sigma, \sigma) \mid \sigma \in supp({\eta_1}^{(State)})\}$ and $\eta_1' = \delta(C; \textbf{while}\ (b)\ \textbf{do}\ C) \otimes {\eta_1}^{(State)}$. From ${\eta_2}^{(Stmt)} = \delta(\textbf{while}\ (b)\ \textbf{do}\ C)$, ${\eta_2}^{(State)} \models \lceil \neg b \rceil$ and $\eta_2 \rightsquigarrow (\theta_1, \eta_1')$ by Lem. 220 and Lem. 193 we know $\theta_2 = \{(\sigma, \sigma) \mid \sigma \in supp({\eta_2}^{(State)})\}$ and $\eta_2' = \delta(C; \textbf{while}\ (b)\ \textbf{do}\ C) \otimes {\eta_2}^{(State)}$, thus $\theta = \theta_1 \cup \theta_2 \subseteq [\![\mathbf{Id}]\!] \subseteq [\![G_1 \vee G_2]\!]$ and $\eta' = \eta_1' \oplus_p \eta_2' = (\delta(C; \textbf{while}\ (b)\ \textbf{do}\ C) \otimes {\eta_1}^{(State)}) \oplus_p (\delta(\textbf{skip}) \otimes {\eta_2}^{(State)})$. By Lem. 12 and Lem. 19 we know ${\eta'}^{(State)} = {\eta_1}^{(State)} \oplus_p {\eta_2}^{(State)} = \eta^{(State)} \models I$. To prove $(\eta', R, I) \Longmapsto^k_{\text{NST}} (G_1 \vee G_2, Q)$, from $G_2 \Rightarrow G_1 \vee G_2$ by Lem. 181, it suffices to prove $(\eta', R, I) \Longmapsto^k_{\text{NST}} (G_2, Q)$. From $supp(\eta_1) \subseteq supp(\eta)$ by Lem. 24 we know $supp({\eta_1}^{(State)}) \subseteq supp(\eta^{(State)})$. From $\eta^{(State)} \models P_2$ and $\mathbf{scl}(P_2)$ we have ${\eta_1}^{(State)} \models P_2$. Similarly we can prove ${\eta_2}^{(State)} \models P_2$. From ${\eta_2}^{(State)} \models \lceil \neg b \rceil$ we know ${\eta_2}^{(State)} \models P_2 \wedge \lceil \neg b \rceil$. From $P_2 \wedge \lceil \neg b \rceil \Rightarrow Q$ we have ${\eta_2}^{(State)} \models Q$.
From $\textbf{disablesplit}(\lceil \mathbf{q} \rceil, C)$ we know
$\textbf{disablesplit}(\lceil \mathbf{q} \rceil, \textbf{while}\ (b)\ \textbf{do}\ C)$, thus $\textbf{disablesplit}(\lceil \mathbf{q} \rceil, C; \textbf{while}\ (b)\ \textbf{do}\ C)$.
By Lem. 286 we know $\textbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C; \textbf{while}\ (b)\ \textbf{do}\ C) \otimes {\eta_1}^{(State)})$. To prove $(\eta', R, I) \Longmapsto^k_{\text{NST}} (G_2, Q)$, from $0 < p < 1$, $\eta' = (\delta(C; \textbf{while}\ (b)\ \textbf{do}\ C) \otimes {\eta_1}^{(State)}) \oplus_p (\delta(\textbf{skip}) \otimes {\eta_2}^{(State)})$, ${\eta'}^{(State)} \models I$, ${\eta_2}^{(State)} \models Q$, $\textbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C; \textbf{while}\ (b)\ \textbf{do}\ C) \otimes {\eta_1}^{(State)})$, $\textbf{closed}(Q)$, $\textbf{sta}(\mathbf{q}, R)$, $\textbf{Sta}(Q, R, \text{true})$, $\mathbf{Id} \Rightarrow G_2$ and $\forall x \in fv(I).\ G_2 \Rightarrow \textbf{Inv}(x)$ by Lem. 302, it suffices to prove $(\delta(C; \textbf{while}\ (b)\ \textbf{do}\ C) \otimes {\eta_1}^{(State)}, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G_2, Q)$. By Lem. 217 we know $\delta(C; \textbf{while}\ (b)\ \textbf{do}\ C) \otimes {\eta_1}^{(State)} = \delta(C) \otimes {\eta_1}^{(State)}; \textbf{while}\ (b)\ \textbf{do}\ C)$, thus we need to prove
$(\delta(C) \otimes {\eta_1}^{(State)}; \textbf{while}\ (b)\ \textbf{do}\ C), R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G_2, Q)$. From $\textbf{disablesplit}(\lceil \mathbf{q} \rceil, C)$ by Lem. 286 we know $\textbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C) \otimes {\eta_1}^{(State)})$. From ${\eta_1}^{(State)} \models P_2$ and ${\eta_1}^{(State)} \models \lceil b \rceil$ we know ${\eta_1}^{(State)} \models P_2 \wedge \lceil b \rceil$. From $R, G_2, \lceil \mathbf{q} \rceil \models_{\text{NST}} \{P_2 \wedge \lceil b \rceil\} C \{P_2\}$ we know $(\delta(C) \otimes {\eta_1}^{(State)}, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G_2, P_2)$. To prove $(\delta(C) \otimes {\eta_1}^{(State)}; \textbf{while}\ (b)\ \textbf{do}\ C, R, \lceil \mathbf{q} \rceil) \Longmapsto^k_{\text{NST}} (G_2, Q)$, from $(\delta(C) \otimes$

$\eta_1{}^{(State)}, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G_2, P_2)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, \delta(C) \otimes \eta_1{}^{(State)})$, $\mathbf{disablesplit}(\mathbf{while}\ (b)\ \mathbf{do}\ C)$, $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{closed}(Q)$ $\mathbf{scl}(P_2)$, $\mathbf{Id} \Rightarrow R$ and $\mathbf{Id} \Rightarrow G_2$ by Lem. 299, it suffices to prove $(\delta(\mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G_2, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P_2$. From $R, G_2, \lceil \mathbf{q} \rceil \models_{\text{NST}} \{P_2 \wedge \lceil b \rceil\} C \{P_2\}$, $P_2 \wedge \lceil \neg b \rceil \Rightarrow Q$, $\mathbf{closed}(Q)$, $\mathbf{Sta}(P_2, R, \text{true})$, $\mathbf{Sta}(Q, R, \text{true})$, $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C)$, $\mathbf{scl}(P_2)$, $\mathbf{Id} \Rightarrow R$ and $\mathbf{Id} \Rightarrow G_2$ by Lem. 300 we know $R, G_2, \lceil \mathbf{q} \rceil \models_{\text{NST}} \{P_2\} \mathbf{while}\ (b)\ \mathbf{do}\ C \{Q\}$, thus $(\delta(\mathbf{while}\ (b)\ \mathbf{do}\ C) \otimes \mu, R, \lceil \mathbf{q} \rceil) \Longmapsto^{k}_{\text{NST}} (G_2, Q)$ for all μ such that $\mu \models \lceil \mathbf{q} \rceil \wedge P_2$.

Lemma 304. *For all C, R, G, I, P, Q, if $R, G, I \vdash_{NST} \{P\} C \{Q\}$, then $R, G, I \models_{NST} \{P\} C \{Q\}$.*

Proof. For all C, R, G, I, P, Q such that $R, G, I \vdash_{\text{NST}} \{P\} C \{Q\}$, we prove $R, G, I \models_{\text{NST}} \{P\} C \{Q\}$ by induction on the derivation of $R, G, I \vdash_{\text{NST}} \{P\} C \{Q\}$.

- case (ST-NST): $R, G, I \vdash_{\text{ST}} \{P\} C \{Q\}$.
  From $R, G, I \vdash_{\text{ST}} \{P\} C \{Q\}$ by Lem. 233 we know $R, G, I \models_{\text{ST}} \{P\} C \{Q\}$.
  By Lem. 180 we know $R, G, I \vdash_{\text{NST}} \{P\} C \{Q\}$.
- case (DISJ): $P = P_1 \vee P_2$, $Q = Q_1 \vee Q_2$, $R, G, I \vdash_{\text{NST}} \{P_1\} C \{Q_1\}$ and $R, G, I \vdash_{\text{NST}} \{P_2\} C \{Q_2\}$.
  From $R, G, I \vdash_{\text{NST}} \{P_1\} C \{Q_1\}$ by induction hypothesis we know $R, G, I \models_{\text{NST}} \{P_1\} C \{Q_1\}$. From $R, G, I \vdash_{\text{NST}} \{P_2\} C \{Q_2\}$ by induction hypothesis we know $R, G, I \models_{\text{NST}} \{P_2\} C \{Q_2\}$. By Lem. 183 we know $R, G, I \vdash_{\text{NST}} \{P_1 \vee P_2\} C \{Q_1 \vee Q_2\}$, i.e., $R, G, I \vdash_{\text{NST}} \{P\} C \{Q\}$.
- case (CONJ): $P = P_1 \wedge P_2$, $Q = Q_1 \wedge Q_2$, $R, G, I \vdash_{\text{NST}} \{P_1\} C \{Q_1\}$ and $R, G, I \vdash_{\text{NST}} \{P_2\} C \{Q_2\}$.
  From $R, G, I \vdash_{\text{NST}} \{P_1\} C \{Q_1\}$ by induction hypothesis we know $R, G, I \models_{\text{NST}} \{P_1\} C \{Q_1\}$. From $R, G, I \vdash_{\text{NST}} \{P_2\} C \{Q_2\}$ by induction hypothesis we know $R, G, I \models_{\text{NST}} \{P_2\} C \{Q_2\}$. By Lem. 185 we know $R, G, I \vdash_{\text{NST}} \{P_1 \wedge P_2\} C \{Q_1 \wedge Q_2\}$, i.e., $R, G, I \vdash_{\text{NST}} \{P\} C \{Q\}$.
- case (CSQ): $P \Rightarrow P_1$, $R \Rightarrow R_1$, $G_1 \Rightarrow G$, $Q_1 \Rightarrow Q$ and $R_1, G_1, I \vdash_{\text{NST}} \{P_1\} C \{Q_1\}$.
  From $R_1, G_1, I \vdash_{\text{NST}} \{P_1\} C \{Q_1\}$ by induction hypothesis we know $R_1, G_1, I \models_{\text{NST}} \{P_1\} C \{Q_1\}$. From $P \Rightarrow P_1$, $R \Rightarrow R_1$, $G_1 \Rightarrow G$ and $Q_1 \Rightarrow Q$ by Lem. 182 we know $R, G, I \models_{\text{NST}} \{P\} C \{Q\}$.
- case (SEQ-ST): $C = C_1; C_2$, $R, G, I \vdash_{\text{ST}} \{P\} C_1 \{M\}$ and $R, G, I \vdash_{\text{NST}} \{M\} C_2 \{Q\}$.
  From $R, G, I \vdash_{\text{ST}} \{P\} C_1 \{M\}$ by Lem. 233 we have $R, G, I \models_{\text{ST}} \{P\} C_1 \{M\}$.
  From $R, G, I \vdash_{\text{NST}} \{M\} C_2 \{Q\}$ by induction hypothesis we have $R, G, I \models_{\text{NST}} \{M\} C_2 \{Q\}$.
  By Lem. 218 we know $R, G, I \models_{\text{NST}} \{P\} C_1; C_2 \{Q\}$, i.e., $R, G, I \models_{\text{NST}} \{P\} C \{Q\}$.
- case (SEQ-NST): $C = C_1; C_2$, $G = G_1 \vee G_2$, $R \vee G_2, G_1, I \vdash_{\text{NST}} \{P\} C_1 \{M\}$, $R, G_2, \text{true} \vdash_{\text{NST}} \{M\} C_2 \{Q\}$, $\mathbf{Nosplit}(C_2)$, $\mathbf{closed}(Q)$, $\mathbf{scl}(M)$ and $\forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x)$.
  From $R \vee G_2, G_1, I \vdash_{\text{NST}} \{P\} C_1 \{M\}$ by induction hypothesis we have $R \vee G_2, G_1, I \models_{\text{NST}} \{P\} C_1 \{M\}$. From $R, G_2, \text{true} \vdash_{\text{NST}} \{M\} C_2 \{Q\}$ by induction hypothesis we have $R, G_2, I \models_{\text{NST}} \{M\} C_2 \{Q\}$. From $\mathbf{Nosplit}(C_2)$, $\mathbf{closed}(Q)$,

$\mathbf{scl}(M)$ and $\forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x)$ by Lem. 284 we know $R, G_1 \vee G_2, I \models_{\text{NST}} \{P\}C_1; C_2\{Q\}$, i.e., $R, G, I \models_{\text{NST}} \{P\}C\{Q\}$.

- case (COND): $C = \mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2$, $P = P_1 \vee P_2$, $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $P_1 \Rightarrow \lceil b \rceil$, $P_2 \Rightarrow \lceil \neg b \rceil$, $R, G, I \vdash_{\text{NST}} \{P_1\}C_1\{Q\}$ and $R, G, I \vdash_{\text{NST}} \{P_2\}C_1\{Q\}$. From $R, G, I \vdash_{\text{NST}} \{P_1\}C_1\{Q\}$ by induction hypothesis we know $R, G, I \models_{\text{NST}} \{P_1\}C_1\{Q\}$. From $R, G, I \vdash_{\text{NST}} \{P_2\}C_1\{Q\}$ by induction hypothesis we know $R, G, I \models_{\text{NST}} \{P_2\}C_2\{Q\}$. From $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $P_1 \Rightarrow \lceil b \rceil$ and $P_2 \Rightarrow \lceil \neg b \rceil$ by Lem. 200 we know $R, G, I \models_{\text{NST}} \{P_1 \vee P_2\}\mathbf{if}\ (b)\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2\{Q\}$, i.e., $R, G, I \models_{\text{NST}} \{P\}C\{Q\}$.
- case (WHILE-ST): $C = \mathbf{while}\ (b)\ \mathbf{do}\ C_1$, $P = P_1 \vee P_2$, $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $\mathbf{Sta}(Q, R, I)$, $P_1 \Rightarrow \lceil b \rceil$, $P_2 \Rightarrow \lceil \neg b \rceil \wedge Q$, $R, G, I \vdash_{\text{ST}} \{P_1\}C_1\{P_1 \vee P_2\}$. From $R, G, I \vdash_{\text{ST}} \{P_1\}C_1\{P_1 \vee P_2\}$ by induction hypothesis we know $R, G, I \models_{\text{ST}} \{P_1\}C_1\{P_1 \vee P_2\}$. From $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $\mathbf{Sta}(Q, R, I)$, $P_1 \Rightarrow \lceil b \rceil$ and $P_2 \Rightarrow \lceil \neg b \rceil \wedge Q$ by Lem. 221 we know $R, G, I \models_{\text{ST}} \{P_1 \vee P_2\}\mathbf{while}\ (b)\ \mathbf{do}\ C_1\{Q\}$, i.e., $R, G, I \models_{\text{ST}} \{P\}C\{Q\}$.
- case (WHILE-NST): $C = \mathbf{while}\ (b)\ \mathbf{do}\ C_1$, $P = P_1 \vee P_2$, $G = G_1 \vee G_2$, $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $\mathbf{Sta}(P_2, R, \text{true})$, $\mathbf{Sta}(Q, R, \text{true})$, $P_1 \Rightarrow \lceil b \rceil$, $P_2 \wedge \lceil \neg b \rceil \Rightarrow Q$, $R, G_1, I \vdash_{\text{ST}} \{P_1\}C_1\{P_1 \vee P_2\}$, $P_2 \wedge \lceil b \rceil \Rightarrow \lceil \mathbf{q} \rceil$, $R, G_2, \lceil \mathbf{q} \rceil \vdash_{\text{NST}} \{P_2 \wedge \lceil b \rceil\}C_1\{P_2\}$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C_1)$, $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{closed}(Q)$, $\mathbf{scl}(P_2)$, and $\forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x)$.
  From $R, G_1, I \vdash_{\text{ST}} \{P_1\}C_1\{P_1 \vee P_2\}$ by Lem. 233 we know $R, G_1, I \models_{\text{ST}} \{P_1\}C_1\{P_1 \vee P_2\}$. From $R, G_2, \lceil \mathbf{q} \rceil \vdash_{\text{NST}} \{P_2 \wedge \lceil b \rceil\}C_1\{P_2\}$ by induction hypothesis we know $R, G_2, \lceil \mathbf{q} \rceil \models_{\text{NST}} \{P_2 \wedge \lceil b \rceil\}C_1\{P_2\}$. From $\mathbf{Sta}(P_1 \vee P_2, R, I)$, $\mathbf{Sta}(P_2, R, \text{true})$, $\mathbf{Sta}(Q, R, \text{true})$, $P_1 \Rightarrow \lceil b \rceil$, $P_2 \wedge \lceil \neg b \rceil \Rightarrow Q$, $P_2 \wedge \lceil b \rceil \Rightarrow \lceil \mathbf{q} \rceil$, $\mathbf{disablesplit}(\lceil \mathbf{q} \rceil, C_1)$, $\mathbf{sta}(\mathbf{q}, R)$, $\mathbf{closed}(Q)$, $\mathbf{scl}(P_2)$ and $\forall x \in fv(I).\ G_2 \Rightarrow \mathbf{Inv}(x)$ by Lem. 303 we know $R, G_1 \vee G_2, I \models_{\text{NST}} \{P_1 \vee P_2\}\mathbf{while}\ (b)\ \mathbf{do}\ C_1\{Q\}$, i.e., $R, G, I \models_{\text{ST}} \{P\}C\{Q\}$.

Lemma 305 (Soundness of (SQ-DISJ) rule). *For all C, G, P_1, P_2, Q_1, Q_2, if $G \models_{SQ} \{P_1\}C\{Q_1\}$ and $G \models_{SQ} \{P_2\}C\{Q_2\}$, then $G \models_{SQ} \{P_1 \vee P_2\}C\{Q_1 \vee Q_2\}$.*

Proof. For all C, G, P_1, P_2, Q_1, Q_2 such that $G \models_{\text{SQ}} \{P_1\}C\{Q_1\}$ and $G \models_{\text{SQ}} \{P_2\}C\{Q_2\}$, to prove $G \models_{\text{SQ}} \{P_1 \vee P_2\}C\{Q_1 \vee Q_2\}$, we need to prove for all μ, if $\mu \models P_1 \vee P_2$ and $|[\![C]\!](\mu)| = 1$, then $[\![C]\!](\mu) \models Q_1 \vee Q_2$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$. For all μ such that $\mu \models P_1 \vee P_2$ and $|[\![C]\!](\mu)| = 1$, from $\mu \models P_1 \vee P_2$ we know $\mu \models P_1$ or $\mu \models P_2$. We prove the two cases respectively.

- $\mu \models P_1$.
  From $G \models_{\text{SQ}} \{P_1\}C\{Q_1\}$, $\mu \models P_1$ and $|[\![C]\!](\mu)| = 1$ we know $[\![C]\!](\mu) \models Q_1$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$. From $[\![C]\!](\mu) \models Q_1$ we know $[\![C]\!](\mu) \models Q_1 \vee Q_2$.
- $\mu \models P_2$.
  From $G \models_{\text{SQ}} \{P_2\}C\{Q_2\}$, $\mu \models P_2$ and $|[\![C]\!](\mu)| = 1$ we know $[\![C]\!](\mu) \models Q_2$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$. From $[\![C]\!](\mu) \models Q_2$ we know $[\![C]\!](\mu) \models Q_1 \vee Q_2$.

Lemma 306 (Soundness of (SQ-CONJ) rule). *For all C, G, P_1, P_2, Q_1, Q_2, if $G \models_{sQ} \{P_1\}C\{Q_1\}$ and $G \models_{sQ} \{P_2\}C\{Q_2\}$, then $G \models_{sQ} \{P_1 \wedge P_2\}C\{Q_1 \wedge Q_2\}$.*

Proof. For all C, G, P_1, P_2, Q_1, Q_2 such that $G \models_{\text{SQ}} \{P_1\}C\{Q_1\}$ and $G \models_{\text{SQ}} \{P_2\}C\{Q_2\}$, to prove $G \models_{\text{SQ}} \{P_1 \wedge P_2\}C\{Q_1 \wedge Q_2\}$, we need to prove for all μ, if $\mu \models P_1 \wedge P_2$ and $|[\![C]\!](\mu)| = 1$, then $[\![C]\!](\mu) \models Q_1 \vee Q_2$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$. For all μ such that $\mu \models P_1 \wedge P_2$ and $|[\![C]\!](\mu)| = 1$, from $\mu \models P_1 \wedge P_2$ we know $\mu \models P_1$ and $\mu \models P_2$. From $G \models_{\text{SQ}} \{P_1\}C\{Q_1\}$, $\mu \models P_1$ and $|[\![C]\!](\mu)| = 1$ we know $[\![C]\!](\mu) \models Q_1$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$. From $G \models_{\text{SQ}} \{P_2\}C\{Q_2\}$, $\mu \models P_2$ and $|[\![C]\!](\mu)| = 1$ we know $[\![C]\!](\mu) \models Q_2$. From $[\![C]\!](\mu) \models Q_1$ and $[\![C]\!](\mu) \models Q_2$ we know $[\![C]\!](\mu) \models Q_1 \wedge Q_2$.

Lemma 307. *For all C, μ, X, r, if $X \notin wv(C)$, then $[\![C]\!](\mu\{X \rightsquigarrow r\}) = [\![C]\!](\mu)\{X \rightsquigarrow r\}$.*

Proof. For all C, σ, X, r such that $X \notin wv(C)$, we have

$$
\begin{aligned}
& [\![C]\!](\mu\{X \rightsquigarrow r\}) \\
&= \mathbb{E}_{\sigma \sim \mu\{X \rightsquigarrow r\}}\{[\![C]\!](\sigma)\} \\
&= \mathbb{E}_{\sigma \sim \mathbb{E}_{\sigma' \sim \mu}\{\delta(\sigma'\{X \rightsquigarrow r\})\}}\{[\![C]\!](\sigma)\} \\
&= \mathbb{E}_{\sigma' \sim \mu}\{\mathbb{E}_{\sigma \sim \delta(\sigma'\{X \rightsquigarrow r\})}\{[\![C]\!](\sigma)\}\} \qquad \text{(by Lem. 15)} \\
&= \mathbb{E}_{\sigma' \sim \mu}\{[\![C]\!](\sigma'\{X \rightsquigarrow r\})\} \qquad \text{(by Lem. 17)} \\
&= \mathbb{E}_{\sigma' \sim \mu}\{[\![C]\!](\sigma')\{X \rightsquigarrow r\}\} \\
&= \mathbb{E}_{\sigma' \sim \mu}\{\mathbb{E}_{\sigma \sim [\![C]\!](\sigma')}\{\delta(\sigma\{X \rightsquigarrow r\})\}\} \\
&= \mathbb{E}_{\sigma \sim \mathbb{E}_{\sigma' \sim \mu}\{[\![C]\!](\sigma')\}}\{\delta(\sigma\{X \rightsquigarrow r\})\} \qquad \text{(by Lem. 15)} \\
&= \mathbb{E}_{\sigma \sim [\![C]\!](\mu)}\{\delta(\sigma\{X \rightsquigarrow r\})\} \\
&= [\![C]\!](\mu)\{X \rightsquigarrow r\}.
\end{aligned}
$$

Lemma 308. *For all $\mu \in \mathbb{SD}_{State}, X, r$, $|\mu\{X \rightsquigarrow r\}| = |\mu|$.*

Proof. For all $\mu \in \mathbb{SD}_{State}, X, r$, we have $|\mu\{X \rightsquigarrow r\}| = \sum_{\sigma'} \mu\{X \rightsquigarrow r\}(\sigma') = \sum_{\sigma'} \sum_{\sigma}\{\mu(\sigma) \mid \sigma\{X \rightsquigarrow r\} = \sigma'\} = \sum_{\sigma} \mu(\sigma) = |\mu|$.

Lemma 309. *For all σ, μ, X, r, if $\sigma \in \mathit{supp}(\mu)$, then $\sigma\{X \rightsquigarrow r\} \in \mathit{supp}(\mu\{X \rightsquigarrow r\})$.*

Proof. For all σ, μ, X, r such that $\sigma \in \mathit{supp}(\mu)$, we know $\mu(\sigma) > 0$, thus $\mu\{X \rightsquigarrow r\}(\sigma\{X \rightsquigarrow r\}) = \sum_{\sigma'}\{\mu(\sigma') \mid \sigma'\{X \rightsquigarrow r\} = \sigma\{X \rightsquigarrow r\}\} \geq \mu(\sigma) > 0$, so $\sigma\{X \rightsquigarrow r\} \in \mathit{supp}(\mu\{X \rightsquigarrow r\})$.

Lemma 310 (Soundness of (SQ-EXISTS) rule). *For all C, P, Q, G, X, if $X \notin fv(G) \cup wv(C)$ and $G \models_{sQ} \{P\}C\{Q\}$, then $G \models_{sQ} \{\exists X.P\}C\{\exists X.Q\}$.*

Proof. For all C, P, Q, G, X such that $X \notin fv(G) \cup wv(C)$ and $G \models_{\text{SQ}} \{P\}C\{Q\}$, from $X \notin fv(G) \cup fv(G)$ we know $X \notin fv(G)$ and $X \notin wv(C)$. To prove $G \models_{\text{SQ}} \{\exists X.P\}C\{\exists X.Q\}$, we need to prove for all μ, if $\mu \models \exists X.P$ and $|[\![C]\!](\mu)| = 1$, then $[\![C]\!](\mu) \models \exists X.Q$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$

and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$. For all μ such that $\mu \models \exists X.P$ and $|[\![C]\!](\mu)| = 1$, from $\mu \models \exists X.P$ we know there exists r such that $\mu\{X \rightsquigarrow r\} \models P$. From $X \notin wv(C)$ by Lem. 307 we know $[\![C]\!](\mu\{X \rightsquigarrow r\}) = [\![C]\!](\mu)\{X \rightsquigarrow r\}$. By Lem. 308 we know $|[\![C]\!](\mu\{X \rightsquigarrow r\})| = |[\![C]\!](\mu)\{X \rightsquigarrow r\}| = |[\![C]\!](\mu)| = 1$. From $G \models_{\text{SQ}} \{P\}C\{Q\}$, $\mu\{X \rightsquigarrow r\} \models P$ and $|[\![C]\!](\mu\{X \rightsquigarrow r\})| = 1$ we know $[\![C]\!](\mu\{X \rightsquigarrow r\}) \models Q$, i.e., $[\![C]\!](\mu)\{X \rightsquigarrow r\} \models Q$, thus $[\![C]\!](\mu) \models \exists X.Q$. For all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$, by Lem. 309 we know $\sigma\{X \rightsquigarrow r\} \in \mathit{supp}(\mu\{X \rightsquigarrow r\})$ and $\sigma'\{X \rightsquigarrow r\} \in \mathit{supp}([\![C]\!](\sigma)\{X \rightsquigarrow r\})$. From $X \notin wv(C)$ we know $[\![C]\!](\sigma)\{X \rightsquigarrow r\} = [\![C]\!](\sigma\{X \rightsquigarrow r\})$, thus $\sigma'\{X \rightsquigarrow r\} \in \mathit{supp}([\![C]\!](\sigma\{X \rightsquigarrow r\}))$. From $G \models_{\text{SQ}} \{P\}C\{Q\}$, $\mu\{X \rightsquigarrow r\} \models P$, $|[\![C]\!](\mu\{X \rightsquigarrow r\})| = 1$, $\sigma\{X \rightsquigarrow r\} \in \mathit{supp}(\mu\{X \rightsquigarrow r\})$ and $\sigma'\{X \rightsquigarrow r\} \in \mathit{supp}([\![C]\!](\sigma\{X \rightsquigarrow r\}))$ we know $(\sigma\{X \rightsquigarrow r\}, \sigma'\{X \rightsquigarrow r\}) \models G$, thus $(\sigma, \sigma') \models \exists X.G$. From $X \notin fv(G)$ we know $(\sigma, \sigma') \models G$.

Lemma 311 (Soundness of (SQ-FORALL) rule). *For all C, P, Q, G, X, if $X \notin fv(G) \cup wv(C)$ and $G \models_{\text{SQ}} \{P\}C\{Q\}$, then $G \models_{\text{SQ}} \{\forall X.P\}C\{\forall X.Q\}$.*

Proof. For all C, P, Q, G, X such that $X \notin fv(G) \cup wv(C)$ and $G \models_{\text{SQ}} \{P\}C\{Q\}$, from $X \notin fv(G) \cup fv(G)$ we know $X \notin fv(G)$ and $X \notin wv(C)$. To prove $G \models_{\text{SQ}} \{\forall X.P\}C\{\forall X.Q\}$, we need to prove for all μ, if $\mu \models \forall X.P$ and $|[\![C]\!](\mu)| = 1$, then $[\![C]\!](\mu) \models \forall X.Q$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$. For all μ such that $\mu \models \forall X.P$ and $|[\![C]\!](\mu)| = 1$, to prove $[\![C]\!](\mu) \models \forall X.Q$, we need to prove $[\![C]\!](\mu)\{X \rightsquigarrow r\} \models Q$ for all r. For all r, from $\mu \models \exists X.P$ we know $\mu\{X \rightsquigarrow r\} \models P$. From $X \notin wv(C)$ by Lem. 307 we know $[\![C]\!](\mu\{X \rightsquigarrow r\}) = [\![C]\!](\mu)\{X \rightsquigarrow r\}$. By Lem. 308 we know $|[\![C]\!](\mu\{X \rightsquigarrow r\})| = |[\![C]\!](\mu)\{X \rightsquigarrow r\}| = |[\![C]\!](\mu)| = 1$. From $G \models_{\text{SQ}} \{P\}C\{Q\}$, $\mu\{X \rightsquigarrow r\} \models P$ and $|[\![C]\!](\mu\{X \rightsquigarrow r\})| = 1$ we know $[\![C]\!](\mu\{X \rightsquigarrow r\}) \models Q$, i.e., $[\![C]\!](\mu)\{X \rightsquigarrow r\} \models Q$. For all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$, by Lem. 309 we know $\sigma\{X \rightsquigarrow 0\} \in \mathit{supp}(\mu\{X \rightsquigarrow 0\})$ and $\sigma'\{X \rightsquigarrow 0\} \in \mathit{supp}([\![C]\!](\sigma)\{X \rightsquigarrow 0\})$. From $X \notin wv(C)$ we know $[\![C]\!](\sigma)\{X \rightsquigarrow 0\} = [\![C]\!](\sigma\{X \rightsquigarrow 0\})$, thus $\sigma'\{X \rightsquigarrow 0\} \in \mathit{supp}([\![C]\!](\sigma\{X \rightsquigarrow 0\}))$. From $\mu \models \forall X.P$ we know $\mu\{X \rightsquigarrow 0\} \models P$. By Lem. 308 we know $|[\![C]\!](\mu\{X \rightsquigarrow 0\})| = |[\![C]\!](\mu)\{X \rightsquigarrow 0\}| = |[\![C]\!](\mu)| = 1$. From $G \models_{\text{SQ}} \{P\}C\{Q\}$, $\mu\{X \rightsquigarrow 0\} \models P$, $|[\![C]\!](\mu\{X \rightsquigarrow 0\})| = 1$, $\sigma\{X \rightsquigarrow 0\} \in \mathit{supp}(\mu\{X \rightsquigarrow 0\})$ and $\sigma'\{X \rightsquigarrow 0\} \in \mathit{supp}([\![C]\!](\sigma\{X \rightsquigarrow 0\}))$ we know $(\sigma\{X \rightsquigarrow 0\}, \sigma'\{X \rightsquigarrow 0\}) \models G$, thus $(\sigma, \sigma') \models \exists X.G$. From $X \notin fv(G)$ we know $(\sigma, \sigma') \models G$.

Lemma 312 (Soundness of (SQ-CSQ) rule). *For all C, P, P', Q, Q', G, G', if $P \Rightarrow P'$, $G' \models_{\text{SQ}} \{P'\}C\{Q'\}$, $Q' \Rightarrow Q$ and $G' \Rightarrow G$, then $G \models_{\text{SQ}} \{P\}C\{Q\}$.*

Proof. For all C, P, P', Q, Q', G, G' such that $P \Rightarrow P'$, $G' \models_{\text{SQ}} \{P'\}C\{Q'\}$, $Q' \Rightarrow Q$ and $G' \Rightarrow G$, to prove $G \models_{\text{SQ}} \{P\}C\{Q\}$, we need to prove for all μ, if $\mu \models P$ and $|[\![C]\!](\mu)| = 1$, then $[\![C]\!](\mu) \models Q$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$. For all μ such that $\mu \models P$ and $|[\![C]\!](\mu)| = 1$, from $\mu \models P$ and $P \Rightarrow P'$ we know $\mu \models P'$. From $G \models_{\text{SQ}} \{P'\}C\{Q'\}$, $\mu \models P'$ and $|[\![C]\!](\mu)| = 1$ we know $[\![C]\!](\mu) \models Q'$ and $(\sigma, \sigma') \models G'$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$. From $Q' \Rightarrow Q$ and $G' \Rightarrow G$ we

know $[\![C]\!](\mu) \models Q$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$.

Lemma 313. *For all* C, μ_1, μ_2, p, $[\![C]\!](\mu_1 \oplus_p \mu_2) = [\![C]\!](\mu_1) \oplus_p [\![C]\!](\mu_2)$.

Proof. For all C, μ_1, μ_2, p, by Lem. 16 we know $[\![C]\!](\mu_1 \oplus_p \mu_2) = \mathbb{E}_{\sigma \sim \mu_1 \oplus_p \mu_2}\{[\![C]\!](\sigma)\} =$

$\mathbb{E}_{\sigma \sim \mu_1}\{[\![C]\!](\sigma)\} \oplus_p \mathbb{E}_{\sigma \sim \mu_2}\{[\![C]\!](\sigma)\} = [\![C]\!](\mu_1) \oplus_p [\![C]\!](\mu_2)$.

Lemma 314. *For all set* A *and* $\mu_1, \mu_2 \in \mathbb{SD}_A$, $p \in (0,1)$, *if* $|\mu_1 \oplus_p \mu_2| = 1$, *then* $|\mu_1| = 1$ *and* $|\mu_2| = 1$.

Proof. For all set A and $\mu_1, \mu_2 \in \mathbb{SD}_A$, $p \in (0,1)$ such that $|\mu_1 \oplus_p \mu_2| = 1$, we have $1 = |\mu_1 \oplus_p \mu_2| = \sum_a (\mu_1(a) \oplus_p \mu(a)) = \sum_a p \cdot \mu_1(a) + (1-p) \cdot \mu_2(a) = p \cdot |\mu_1| + (1-p) \cdot |\mu_2|$. From $0 < p < 1$ we know $|\mu_1| = 1$ and $|\mu_2| = 1$.

Lemma 315 (Soundness of (SQ-OPLUS) rule). *For all* $C, G, P_1, P_2, Q_1, Q_2, p$, *if* $G \models_{sQ} \{P_1\}C\{Q_1\}$ *and* $G \models_{sQ} \{P_2\}C\{Q_2\}$, *then* $G \models_{sQ} \{P_1 \oplus_p P_2\}C\{Q_1 \oplus_p Q_2\}$

Proof. For all $C, G, P_1, P_2, Q_1, Q_2, p$ such that $G \models_{\text{SQ}} \{P_1\}C\{Q_1\}$ and $G \models_{\text{SQ}} \{P_2\}C\{Q_2\}$, there are three cases: $p = 0$, $p = 1$ or $0 < p < 1$. The cases $p = 0$ and $p = 1$ are trivial. We only prove the case $0 < p < 1$. To prove $G \models_{\text{SQ}} \{P_1 \oplus_p P_2\}C\{Q_1 \oplus_p Q_2\}$, we need to prove for all μ, if $\mu \models P_1 \oplus_p P_2$ and $|[\![C]\!](\mu)| = 1$, then $[\![C]\!](\mu) \models Q_1 \oplus_p Q_2$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$. For all μ such that $\mu \models P_1 \oplus_p P_2$ and $|[\![C]\!](\mu)| = 1$, from $\mu \models P_1 \oplus_p P_2$ and $0 < p < 1$ we know there exists μ_1 and μ_2 such that $\mu = \mu_1 \oplus_p \mu_2$, $\mu_1 \models P_1$ and $\mu_2 \models P_2$. By Lem. 313 we know $[\![C]\!](\mu) = [\![C]\!](\mu_1 \oplus_p \mu_2) = [\![C]\!](\mu_1) \oplus_p [\![C]\!](\mu_2)$. From $0 < p < 1$ by Lem. 314 we know $|[\![C]\!](\mu_1)| = 1$ and $|[\![C]\!](\mu_2)| = 1$. From $G \models_{\text{SQ}} \{P_1\}C\{Q_1\}$, $\mu_1 \models P_1$ and $|[\![C]\!](\mu_1)| = 1$ we know $[\![C]\!](\mu_1) \models Q_1$. From $G \models_{\text{SQ}} \{P_2\}C\{Q_2\}$, $\mu_2 \models P_2$ and $|[\![C]\!](\mu_2)| = 1$ we know $[\![C]\!](\mu_2) \models Q_2$. From $[\![C]\!](\mu_1) \models Q_1$, $[\![C]\!](\mu_2) \models Q_2$, $0 < p < 1$ and $[\![C]\!](\mu) = [\![C]\!](\mu_1) \oplus_p [\![C]\!](\mu_2)$ we know $[\![C]\!](\mu) \models Q_1 \oplus_p Q_2$. For all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$, from $0 < p < 1$ by Lem. 275 we know $\mathit{supp}(\mu) = \mathit{supp}(\mu_1 \oplus_p \mu_2) = \mathit{supp}(\mu_1) \cup \mathit{supp}(\mu_2)$. From $\sigma \in \mathit{supp}(\mu)$ we know $\sigma \in \mathit{supp}(\mu_1)$ or $\sigma \in \mathit{supp}(\mu_2)$. If $\sigma \in \mathit{supp}(\mu_1)$, from $G \models_{\text{SQ}} \{P_1\}C\{Q_1\}$, $\mu_1 \models P_1$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$ we know $(\sigma, \sigma') \models G$. If $\sigma \in \mathit{supp}(\mu_2)$, from $G \models_{\text{SQ}} \{P_2\}C\{Q_2\}$, $\mu_2 \models P_2$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$ we know $(\sigma, \sigma') \models G$.

Definition 87. Let $V \in \mathbb{D}_{\mathbb{D}_{State}}$, we define $[\![C]\!](V) \stackrel{\text{def}}{=} \lambda\mu.\ \mathbf{Pr}_{\nu \sim V}[[\![C]\!](\nu) = \mu]$.

Lemma 316. *For all* $V, V' \in \mathbb{D}_{\mathbb{D}_{State}}$ *and* C, $[\![C]\!](\overline{V}) = \overline{[\![C]\!](V)}$.

Proof. For all $V, V' \in \mathbb{D}_{\mathbb{D}_{State}}$ and C, we have

$$\begin{aligned}
[\![C]\!](\overline{V}) &= \mathbb{E}_{\sigma\sim\overline{V}}\{[\![C]\!](\sigma)\} \\
&= \mathbb{E}_{\sigma\sim\mathbb{E}_{\nu\sim V}\{\nu\}}\{[\![C]\!](\sigma)\} \\
&= \mathbb{E}_{\nu\sim V}\{\mathbb{E}_{\sigma\sim\nu}\{[\![C]\!](\sigma)\}\} \qquad \text{(by Lem. 15)} \\
&= \mathbb{E}_{\nu\sim V}\{[\![C]\!]_\nu\} \\
&= \lambda\sigma.\sum_\nu V(\nu)\cdot[\![C]\!]_\nu(\sigma) \\
&= \lambda\sigma.\sum_{\mu,\nu}\{V(\nu)\cdot\mu(\sigma) \mid [\![C]\!]_\nu = \mu\} \\
&= \lambda\sigma.\sum_\mu \mu(\sigma)\cdot\sum_\nu\{V(\nu)\cdot \mid [\![C]\!]_\nu = \mu\} \\
&= \lambda\sigma.\sum_\mu \mu(\sigma)\cdot\mathbf{Pr}_{\nu\sim V}[[\![C]\!]_\nu = \mu] \\
&= \lambda\sigma.\sum_\mu \mu(\sigma)\cdot[\![C]\!](V)(\mu) \\
&= \overline{[\![C]\!](V)}.
\end{aligned}$$

Lemma 317. *For all set A and $V \in \mathbb{SD}_{\mathbb{SD}_A}$, if $|\overline{V}| = 1$, then $|\nu| = 1$ for all $\nu \in supp(V)$.*

Proof. For all set A and $V \in \mathbb{SD}_{\mathbb{SD}_A}$ such that $|\overline{V}| = 1$, we prove by contradiction. Assume there exists $\mu \in supp(V)$ such that $|\mu| \neq 1$, then $V(\mu) > 0$ and $|\mu| < 1$, so $V(\mu)\cdot|\mu| < V(\mu)$, thus $|\overline{V}| = \sum_\sigma \overline{V}(\sigma) = \sum_\sigma\sum_\nu V(\nu)\cdot\nu(\sigma) = \sum_\nu V(\nu)\cdot\sum_\sigma\nu(\sigma) = \sum_\nu V(\nu)\cdot|\nu| = V(\mu)\cdot|\mu| + \sum_\nu\{V(\nu)\cdot|\nu| \mid \nu \neq \mu\} < V(\mu) + \sum_\nu\{V(\nu) \mid \nu \neq \mu\} = \sum_\nu V(\nu) = |V| \leq 1$, which contradicts with $|\overline{V}| = 1$.

Lemma 318. *For all $V \in \mathbb{D}_{\mathbb{D}_{State}}$ and μ, if $\mu \in supp(V)$, then $[\![C]\!](\mu) \in supp([\![C]\!](V))$.*

Proof. For all $V \in \mathbb{D}_{\mathbb{D}_{State}}$ and μ such that $\mu \in supp(V)$, we know $V(\mu) > 0$, thus $[\![C]\!](V)([\![C]\!](\mu)) = \mathbf{Pr}_{\nu\sim V}[[\![C]\!](\nu) = [\![C]\!](\mu)] \geq V(\mu) > 0$, so $[\![C]\!](\mu) \in supp([\![C]\!](V))$.

Lemma 319. *For all set A and $\mu \in \mathbb{D}_{\mathbb{D}_A}$, $a \in supp(\overline{\mu})$, there exists $\nu \in supp(\mu)$ such that $a \in supp(\nu)$.*

Proof. For all set A and $\mu \in \mathbb{D}_{\mathbb{D}_A}$, $a \in supp(\overline{\mu})$, we know $0 < \overline{\mu}(a) = \sum_{\nu\in\mathbb{D}_A}\mu(\nu)\cdot\nu(a)$, so there exists ν such that $\mu(\nu) > 0$ and $\nu(a) > 0$, thus $\nu \in supp(\mu)$ and $a \in supp(\nu)$.

Lemma 320 (Soundness of (SQ-BIGOPLUS) rule). *For all C, G, P, Q, if $G \models_{SQ} \{P\}C\{Q\}$, then $G \models_{SQ} \{\bigoplus P\}C\{\bigoplus Q\}$.*

Proof. For all C, G, P, Q such that $G \models_{SQ} \{P\}C\{Q\}$, to prove $G \models_{SQ} \{\bigoplus P\}C\{\bigoplus Q\}$, we need to prove for all μ, if $\mu \models \bigoplus P$ and $|[\![C]\!](\mu)| = 1$, then $[\![C]\!](\mu) \models \bigoplus Q$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in supp(\mu)$ and $\sigma' \in supp([\![C]\!](\sigma))$. For all μ such that $\mu \models \bigoplus P$ and $|[\![C]\!](\mu)| = 1$, from $\mu \models \bigoplus P$ we know there exists $V \in \mathbb{D}_{\mathbb{D}_{State}}$ such that $\mu = \overline{V}$ and $\nu \models P$ for all $\nu \in supp(V)$. By Lem. 316 we know $[\![C]\!](\mu) = [\![C]\!](\overline{V}) = \overline{[\![C]\!](V)}$. From $|[\![C]\!](\mu)| = 1$ we know $|\overline{[\![C]\!](V)}| = 1$. For all $\mu' \in supp([\![C]\!](V))$, by Lem. 317 we know $|\mu'| = 1$. From $\mu' \in supp([\![C]\!](V))$ we know $0 < [\![C]\!](V)(\mu') = \mathbf{Pr}_{\nu\sim V}[[\![C]\!](\nu) = \mu'] = \sum_\nu\{V(\nu) \mid [\![C]\!](\nu) = \mu'\}$, so there exists ν such that $\nu \in supp(V)$ and $[\![C]\!](\nu) = \mu'$. From $\nu \in supp(V)$ we

know $\nu \models P$. From $|\mu'| = 1$ we know $|[\![C]\!](\nu)| = 1$. From $G \models_{\text{SQ}} \{P\}C\{Q\}$, $\nu \models P$ and $|[\![C]\!](\nu)| = 1$ we know $[\![C]\!](\nu) \models Q$, i.e., $\mu' \models Q$. Therefore, $\mu' \models Q$ for all $\mu' \in supp([\![C]\!](V))$. From $\mathbf{closed}(Q)$ we know $\overline{[\![C]\!](V)} \models Q$. From $[\![C]\!](\mu) = \overline{[\![C]\!](V)}$ we know $[\![C]\!](\mu) \models Q$. For all σ and σ' such that $\sigma \in supp(\mu)$ and $\sigma' \in supp([\![C]\!](\sigma))$, from $\mu = \overline{V}$ we know $\sigma \in supp(\overline{V})$, by Lem. 319 there exists $\nu \in supp(V)$ such that $\nu \in supp(V)$. From $\nu \in supp(V)$ we know $\nu \models P$. From $\nu \in supp(V)$ by Lem. 318 we know $[\![C]\!](\nu) \in supp([\![C]\!](V))$. From $|[\![C]\!](V)| = 1$ by Lem. 317 we know $|[\![C]\!](\nu)| = 1$. From $G \models_{\text{SQ}} \{P\}C\{Q\}$, $\nu \models P$, $|[\![C]\!](\nu)| = 1$, $\sigma \in supp(\nu)$ and $\sigma' \in supp([\![C]\!](\nu))$ we know $(\sigma, \sigma') \models G$.

Lemma 321. *For all n and σ, $(\boldsymbol{skip}, \sigma) \xrightarrow{1}{}^{n} (\boldsymbol{skip}, \sigma)$.*

Proof. by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all σ, $(\mathbf{skip}, \sigma) \xrightarrow{1}{}^{k} (\mathbf{skip}, \sigma)$.
  For all σ, by IH we know $\sum_{C',\sigma'} \{p_1 \cdot p_2 \mid (\mathbf{skip}, \sigma) \xrightarrow{p_1} (C', \sigma') \wedge (C', \sigma') \xrightarrow{p_2}{}^{k} (\mathbf{skip}, \sigma)\} = \sum \{p_2 \mid (\mathbf{skip}, \sigma) \xrightarrow{p_2}{}^{k} (\mathbf{skip}, \sigma)\} = 1$, thus $(\mathbf{skip}, \sigma) \xrightarrow{1}{}^{n} (\mathbf{skip}, \sigma)$.

Lemma 322. *For all n, σ, σ', if $\sigma \neq \sigma'$, then $(\boldsymbol{skip}, \sigma) \xrightarrow{0}{}^{n} (\boldsymbol{skip}, \sigma')$.*

Proof. by induction on n.

- base case: $n = 0$. trivial.
- inductive case: $n = k + 1$.
  IH: for all σ, σ', if $\sigma \neq \sigma'$, then $(\mathbf{skip}, \sigma) \xrightarrow{0}{}^{k} (\mathbf{skip}, \sigma')$.
  For all σ, by IH we know $\sum_{C'',\sigma''} \{p_1 \cdot p_2 \mid (\mathbf{skip}, \sigma) \xrightarrow{p_1} (C'', \sigma'') \wedge (C'', \sigma'') \xrightarrow{p_2}{}^{k} (\mathbf{skip}, \sigma)\} = \sum \{p_2 \mid (\mathbf{skip}, \sigma) \xrightarrow{p_2}{}^{k} (\mathbf{skip}, \sigma')\} = 0$, thus $(\mathbf{skip}, \sigma) \xrightarrow{0}{}^{n} (\mathbf{skip}, \sigma')$.

Lemma 323. *For all σ, $[\![\boldsymbol{skip}]\!](\sigma) = \delta(\sigma)$.*

Proof. For all σ, we have

$$
\begin{aligned}
[\![\mathbf{skip}]\!](\sigma) &= \lambda\sigma'.\lim \vec{p}, \text{where } \forall n.\ (\mathbf{skip}, \sigma) \xrightarrow{\vec{p}\,[n]}{}^{n} (\mathbf{skip}, \sigma') \\
&= \begin{cases} \lim \vec{1}, & \text{if } \sigma' = \sigma \\ \lim \vec{0}, & \text{otherwise} \end{cases} \qquad \text{(by Lem. 321 and Lem. 322)} \\
&= \begin{cases} 1, & \text{if } \sigma' = \sigma \\ 0, & \text{otherwise} \end{cases} \\
&= \delta(\sigma).
\end{aligned}
$$

Lemma 324. *For all μ, $[\![\boldsymbol{skip}]\!](\mu) = \mu$.*

Proof. For all μ, by Lem. 323 and Lem. 17 we know $[\![\mathbf{skip}]\!](\mu) = \mathbb{E}_{\sigma\sim\mu}\{[\![\mathbf{skip}]\!](\sigma)\} = \mathbb{E}_{\sigma\sim\mu}\{\delta(\sigma)\} = \mu$.

Lemma 325 (Soundness of (SQ-SKIP) rule). *For all Q,* $\mathbf{Id} \models_{SQ} \{Q\}\boldsymbol{skip}\{Q\}$.

Proof. For all Q, to prove $\mathbf{Id} \models_{\text{SQ}} \{Q\}\mathbf{skip}\{Q\}$, we need to prove for all μ, if $\mu \models Q$ and $|[\![\mathbf{skip}]\!](\mu)| = 1$, then $[\![\mathbf{skip}]\!](\mu) \models Q$ and $(\sigma, \sigma') \models \mathbf{Id}$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![\mathbf{skip}]\!](\sigma))$. For all μ such that $\mu \models Q$ and $|[\![\mathbf{skip}]\!](\mu)| = 1$, by Lem. 324 we know $[\![\mathbf{skip}]\!](\mu) = \mu \models Q$. For all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![\mathbf{skip}]\!](\sigma))$, by Lem. 323 we know $[\![\mathbf{skip}]\!](\sigma) = \delta(\sigma)$. From $\sigma' \in \mathit{supp}([\![\mathbf{skip}]\!](\sigma))$ we know $\sigma' = \sigma$, thus $(\sigma, \sigma') \models \mathbf{Id}$.

Lemma 326. *For all C_1, C_2, μ, $[\![C_1; C_2]\!](\mu) = [\![C_2]\!]([\![C_1]\!](\mu))$.*

Proof. For all C_1, C_2, μ, we have

$$\begin{aligned}
[\![C_1; C_2]\!](\mu) &= \mathbb{E}_{\sigma\sim\mu}\{[\![C_1; C_2]\!](\sigma)\} \\
&= \mathbb{E}_{\sigma\sim\mu}\{[\![C_2]\!]([\![C_1]\!](\sigma))\} \\
&= \mathbb{E}_{\sigma\sim\mu}\{\mathbb{E}_{\sigma'\sim[\![C_1]\!](\sigma)}[\![C_2]\!](\sigma')\} \\
&= \mathbb{E}_{\sigma'\sim\mathbb{E}_{\sigma\sim\mu}\{[\![C_1]\!](\sigma)\}}\{[\![C_2]\!](\sigma')\} \qquad \text{(by Lem. 15)} \\
&= \mathbb{E}_{\sigma'\sim[\![C_1]\!](\mu)}\{[\![C_2]\!](\sigma')\} \\
&= [\![C_2]\!]([\![C_1]\!](\mu)).
\end{aligned}$$

Lemma 327. *For all $\mu \in \mathbb{SD}_{State}$ and C, if $|[\![C]\!](\mu)| = 1$, then $|\mu| = 1$.*

Proof. For all $\mu \in \mathbb{SD}_{State}$ and C such that $|[\![C]\!](\mu)| = 1$, we know

$$\begin{aligned}
1 &= |[\![C]\!](\mu)| \\
&= \textstyle\sum_{\sigma'} \mathbb{E}_{\sigma\sim\mu}\{[\![C]\!](\sigma)\}(\sigma') \\
&= \textstyle\sum_{\sigma'} \sum_{\sigma} \mu(\sigma) \cdot [\![C]\!](\sigma)(\sigma') \\
&= \textstyle\sum_{\sigma} \mu(\sigma) \cdot \sum_{\sigma'} [\![C]\!](\sigma)(\sigma') \\
&= \textstyle\sum_{\sigma} \mu(\sigma) \cdot |[\![C]\!](\sigma)| \\
&\leq \textstyle\sum_{\sigma} \mu(\sigma) \\
&= |\mu|.
\end{aligned}$$

From $\mu \in \mathbb{SD}_{State}$ we know $|\mu| \leq 1$, thus $|\mu| = 1$.

Lemma 328. *For all C, μ, σ, if $\sigma \in \mathit{supp}([\![C]\!](\mu))$, then there exists σ_0 such that $\sigma_0 \in \mathit{supp}(\mu)$ and $\sigma \in \mathit{supp}([\![C]\!](\sigma_0))$.*

Proof. For all C, μ, σ such that $\sigma \in \mathit{supp}([\![C]\!](\mu))$, we have $0 < [\![C]\!](\mu)(\sigma) = \mathbb{E}_{\sigma_0\sim\mu}\{[\![C]\!](\sigma_0)\}(\sigma) = \sum_{\sigma_0} \mu(\sigma_0) \cdot [\![C]\!](\sigma_0)(\sigma)$, so there exists σ_0 such that $\mu(\sigma_0) > 0$ and $[\![C]\!](\sigma_0)(\sigma) > 0$, thus $\sigma_0 \in \mathit{supp}(\mu)$ and $\sigma \in \mathit{supp}([\![C]\!](\sigma_0))$.

Lemma 329. *For all C, μ, σ, σ', if $\sigma \in \mathit{supp}(\mu)$, then $\mathit{supp}([\![C]\!](\sigma)) \subseteq \mathit{supp}([\![C]\!](\mu))$.*

Proof. For all C, μ, σ, σ' such that $\sigma \in \mathit{supp}(\mu)$, we know $\mu(\sigma) > 0$, to prove $\mathit{supp}([\![C]\!](\sigma)) \subseteq \mathit{supp}([\![C]\!](\mu))$, we need to prove $\sigma' \in \mathit{supp}([\![C]\!](\mu))$ for all $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$. For all $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$, we have $[\![C]\!](\sigma)(\sigma') > 0$, thus $[\![C]\!](\mu)(\sigma') = \mathbb{E}_{\sigma_0\sim\mu}\{[\![C]\!](\sigma_0)\}(\sigma') = \sum_{\sigma_0} \mu(\sigma_0) \cdot [\![C]\!](\sigma_0)(\sigma') \geq \mu(\sigma) \cdot [\![C]\!](\sigma)(\sigma') > 0$, so $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$.

Lemma 330 (Soundness of (SQ-SEQ) rule). *For all* $C_1, C_2, P, M, Q, G_1, G_2$, *if* $G_1 \models_{sQ} \{P\}C_1\{M\}$ *and* $G_2 \models_{sQ} \{M\}C_2\{Q\}$, *then* $G_1 \circ G_2 \models_{sQ} \{P\}C_1; C_2\{Q\}$.

Proof. For all $C_1, C_2, P, M, Q, G_1, G_2$ such that $G_1 \models_{sQ} \{P\}C_1\{M\}$ and $G_2 \models_{sQ} \{M\}C_2\{Q\}$, to prove $G_1 \circ G_2 \models_{sQ} \{P\}C_1; C_2\{Q\}$, we need to prove for all μ, if $\mu \models P$ and $|[\![C_1; C_2]\!](\mu)| = 1$, then $[\![C_1; C_2]\!](\mu) \models Q$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C_1; C_2]\!](\sigma))$. For all μ such that $\mu \models P$ and $|[\![C_1; C_2]\!](\mu)| = 1$, by Lem. 326 we know $[\![C_1; C_2]\!](\mu) = [\![C_2]\!]([\![C_1]\!](\mu))$. From $|[\![C_1; C_2]\!](\mu)| = 1$ we know $|[\![C_2]\!]([\![C_1]\!](\mu)| = 1$. By Lem. 327 we know $|[\![C_1]\!](\mu)| = 1$. From $G_1 \models_{sQ} \{P\}C_1\{M\}$, $\mu \models P$ and $|[\![C_1]\!](\mu)| = 1$ we know $[\![C_1]\!](\mu) \models M$. From $G_2 \models_{sQ} \{M\}C_2\{Q\}$, $[\![C_1]\!](\mu) \models M$ and $|[\![C_2]\!]([\![C_1]\!](\mu)| = 1$ we know $[\![C_2]\!]([\![C_1]\!](\mu) \models Q$. For all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C_1; C_2]\!](\sigma))$, we have $[\![C_1; C_2]\!](\sigma) = [\![C_2]\!]([\![C_1]\!](\sigma))$. From $\sigma' \in \mathit{supp}([\![C_1; C_2]\!](\sigma))$ we know $\sigma' \in \mathit{supp}([\![C_2]\!]([\![C_1]\!](\sigma)))$. By Lem. 328 there exists σ'' such that $\sigma'' \in \mathit{supp}([\![C_1]\!](\sigma))$ and $\sigma' \in \mathit{supp}([\![C_2]\!](\sigma''))$. From $G_1 \models_{sQ} \{P\}C_1\{M\}$, $\mu \models P$, $|[\![C_1]\!](\mu)| = 1$, $\sigma \in \mathit{supp}(\mu)$ and $\sigma'' \in \mathit{supp}([\![C_1]\!](\sigma))$ we know $(\sigma, \sigma'') \models G_1$. From $\sigma \in \mathit{supp}(\mu)$ by Lem. 329 we know $\mathit{supp}([\![C_1]\!](\sigma)) \subseteq \mathit{supp}([\![C_1]\!](\mu)$. From $\sigma'' \in \mathit{supp}([\![C_1]\!](\sigma))$ we know $\sigma'' \in \mathit{supp}([\![C_1]\!](\mu))$. From $G_2 \models_{sQ} \{M\}C_2\{Q\}$, $[\![C_1]\!](\mu) \models M$, $|[\![C_2]\!]([\![C_1]\!](\mu)| = 1$, $\sigma'' \in \mathit{supp}([\![C_1]\!](\mu))$ and $\sigma' \in \mathit{supp}([\![C_2]\!](\sigma''))$ we know $(\sigma'', \sigma') \models G_2$. From $(\sigma, \sigma'') \models G_1$ we know $(\sigma, \sigma') \models G_1 \circ G_2$.

Lemma 331. *For all* x, e, σ, n, *if* $n \geq 1$, *then* $(x := e, \sigma) \xrightarrow{1}{}^{n} (\boldsymbol{skip}, \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})$.

Proof. For all x, e, σ, n such that $n \geq 1$, we have

$$\begin{aligned}
&\textstyle\sum_{C', \sigma'} \{p_1 \cdot p_2 \mid (x := e, \sigma) \xrightarrow{p_1} (C', \sigma') \wedge (C', \sigma') \xrightarrow{p_2}{}^{n-1} (\mathbf{skip}, \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})\} \\
&= \textstyle\sum \{p_2 \mid (\mathbf{skip}, \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}) \xrightarrow{p_2}{}^{n-1} (\mathbf{skip}, \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})\} \\
&= 1. \qquad \text{(by Lem. 321)}
\end{aligned}$$

Therefore $(x := e, \sigma) \xrightarrow{1}{}^{n} (\mathbf{skip}, \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})$.

Lemma 332. *For all* x, e, σ, n, *if* $n \geq 1$ *and* $\sigma' \neq \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}$, *then* $(x := e, \sigma) \xrightarrow{0}{}^{n} (\boldsymbol{skip}, \sigma')$.

Proof. For all x, e, σ, n such that $n \geq 1$, we have

$$\begin{aligned}
&\textstyle\sum_{C', \sigma'} \{p_1 \cdot p_2 \mid (x := e, \sigma) \xrightarrow{p_1} (C', \sigma') \wedge (C', \sigma') \xrightarrow{p_2}{}^{n-1} (\mathbf{skip}, \sigma')\} \\
&= \textstyle\sum \{p_2 \mid (\mathbf{skip}, \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}) \xrightarrow{p_2}{}^{n-1} (\mathbf{skip}, \sigma')\} \\
&= 0. \qquad \text{(by Lem. 322)}
\end{aligned}$$

Therefore $(x := e, \sigma) \xrightarrow{0}{}^{n} (\mathbf{skip}, \sigma')$.

Lemma 333. *For all* σ, x, e, $[\![x := e]\!](\sigma) = \delta(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})$.

Proof. For all σ, x, e, we have

$$\begin{aligned}
[\![x := e]\!](\sigma) &= \lambda\sigma'.\lim \vec{p}, \text{where } \forall n.\ (x := e, \sigma) \xrightarrow{\vec{p}[n]}{}^{n} (\mathbf{skip}, \sigma') \\
&= \begin{cases} \lim\ (0 :: \vec{1}), & \text{if } \sigma' = \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \\ \lim\ (0 :: \vec{0}), & \text{otherwise} \end{cases} \qquad \text{(by Lem. 331 and Lem. 332)} \\
&= \begin{cases} 1, & \text{if } \sigma' = \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \\ 0, & \text{otherwise} \end{cases} \\
&= \delta(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}).
\end{aligned}$$

Definition 88. $\mu\{x \rightsquigarrow e\} \stackrel{\text{def}}{=} \mathbb{E}_{\sigma\sim\mu}\{\delta(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})\}$.

Lemma 334. *For all* μ, x, e, $\mu\{x \rightsquigarrow e\} = \lambda\sigma'.\sum_\sigma\{\mu(\sigma) \mid \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} = \sigma'\}$.

Proof. For all μ, x, e, we have

$$\begin{aligned}
\mu\{x \rightsquigarrow e\} &= \mathbb{E}_{\sigma\sim\mu}\{\delta(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})\} \\
&= \lambda\sigma'.\sum_\sigma \mu(\sigma) \cdot \delta(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})(\sigma') \\
&= \lambda\sigma'.\sum_\sigma\{\mu(\sigma) \mid \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} = \sigma'\}
\end{aligned}$$

Lemma 335. *For all* μ, x, e, $[\![x := e]\!](\mu) = \mu\{x \rightsquigarrow e\}$.

Proof. For all μ, x, e, by Lem. 333 we know $[\![x := e]\!](\mu) = \mathbb{E}_{\sigma\sim\mu}\{[\![x := e]\!](\sigma)\} = \mathbb{E}_{\sigma\sim\mu}\{\delta(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})\} = \mu\{x \rightsquigarrow e\}$.

Lemma 336. *For all* e, x, e', σ, $[\![e[e'/x]]\!]_\sigma = [\![e]\!]_{\sigma\{x\rightsquigarrow[\![e']\!]_\sigma\}}$.

Proof. For all e, x, e', σ, we prove $[\![e[e'/x]]\!]_\sigma = [\![e]\!]_{\sigma\{x\rightsquigarrow[\![e']\!]_\sigma\}}$ by induction on e.

- case n.
  $[\![n[e'/x]]\!]_\sigma = [\![n]\!]_\sigma = n = [\![n]\!]_{\sigma\{x\rightsquigarrow[\![e']\!]_\sigma\}}$.
- case y.
  If $x = y$, then $[\![y[e'/x]]\!]_\sigma = [\![x[e'/x]]\!]_\sigma = [\![e']\!]_\sigma = \sigma\{x \rightsquigarrow [\![e']\!]_\sigma\}(x) = [\![x]\!]_{\sigma\{x\rightsquigarrow[\![e']\!]_\sigma\}} = [\![y]\!]_{\sigma\{x\rightsquigarrow[\![e']\!]_\sigma\}}$. Otherwise $x \neq y$, then $[\![y[e'/x]]\!]_\sigma = [\![y]\!]_\sigma = \sigma(y) = \sigma\{x \rightsquigarrow [\![e']\!]_\sigma\}(y) = [\![y]\!]_{\sigma\{x\rightsquigarrow[\![e']\!]_\sigma\}}$.
- case $e_1 + e_2$.
  IH1: $[\![e_1[e'/x]]\!]_\sigma = [\![e_1]\!]_{\sigma\{x\rightsquigarrow[\![e']\!]_\sigma\}}$.
  IH2: $[\![e_2[e'/x]]\!]_\sigma = [\![e_2]\!]_{\sigma\{x\rightsquigarrow[\![e']\!]_\sigma\}}$.
  By IH1 and IH2 we have $[\![(e_1 + e_2)[e'/x]]\!]_\sigma = [\![e_1[e'/x]]\!]_\sigma + [\![e_2[e'/x]]\!]_\sigma = [\![e_1]\!]_{\sigma\{x\rightsquigarrow[\![e']\!]_\sigma\}} + [\![e_2]\!]_{\sigma\{x\rightsquigarrow[\![e']\!]_\sigma\}} = [\![e_1 + e_2]\!]_{\sigma\{x\rightsquigarrow[\![e']\!]_\sigma\}}$.
- case $e_1 - e_2$.
  Similar to the case $e_1 + e_2$.
- case $e_1 * e_2$.
  Similar to the case $e_1 + e_2$.

Lemma 337. *For all* b, x, e, σ, $[\![b[e/x]]\!]_\sigma = [\![b]\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}}$.

Proof. by induction on b.

– case true.
$[\![\mathrm{true}[e/x]]\!]_\sigma = \{\mathrm{true}\}_\sigma = \mathrm{tt} = [\![\mathrm{true}]\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}}$.
– case false.
$[\![\mathrm{false}[e/x]]\!]_\sigma = \{\mathrm{false}\}_\sigma = \mathrm{ff} = [\![\mathrm{false}]\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}}$.
– case $e_1 < e_2$.

$$\begin{aligned}
[\![(e_1 < e_2)[e/x]]\!]_\sigma &= [\![e_1[e/x] < e_2[e/x]]\!]_\sigma \\
&= \begin{cases} \mathrm{tt}, & \text{if } [\![e_1[e/x]]\!]_\sigma < [\![e_2[e/x]]\!]_\sigma \\ \mathrm{ff}, & \text{otherwise} \end{cases} \\
&= \begin{cases} \mathrm{tt}, & \text{if } [\![e_1]\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}} < [\![e_2]\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}} \\ \mathrm{ff}, & \text{otherwise} \end{cases} \qquad \text{(by Lem. 336)} \\
&= [\![e_1 < e_2]\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}}.
\end{aligned}$$

– case $e_1 = e_2$.
Similar to the case $e_1 < e_2$.
– case $e_1 \le e_2$.
Similar to the case $e_1 < e_2$.
– case $\neg b$.
IH: for all x, e, σ, $[\![b[e/x]]\!]_\sigma = [\![b]\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}}$.
For all x, e, σ, we have

$$\begin{aligned}
[\![(\neg b)[e/x]]\!]_\sigma &= [\![\neg b[e/x]]\!]_\sigma \\
&= \begin{cases} \mathrm{ff}, & \text{if } [\![b[e/x]]\!]_\sigma = \mathrm{tt} \\ \mathrm{tt}, & \text{if } [\![b[e/x]]\!]_\sigma = \mathrm{ff} \end{cases} \\
&= \begin{cases} \mathrm{ff}, & \text{if } [\![b]\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}} = \mathrm{tt} \\ \mathrm{tt}, & \text{if } [\![b]\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}} = \mathrm{ff} \end{cases} \qquad \text{(by IH)} \\
&= [\![b]\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}}.
\end{aligned}$$

– case $b_1 \wedge b_2$.
IH1: for all x, e, σ, $[\![b_1[e/x]]\!]_\sigma = [\![b_1]\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}}$.
IH2: for all x, e, σ, $[\![b_1[e/x]]\!]_\sigma = [\![b_1]\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}}$.
For all x, e, σ, we have

$$\begin{aligned}
&[\![(b_1 \wedge b_2)[e/x]]\!]_\sigma \\
&= [\![b_1[e/x] \wedge b_2[e/x]]\!]_\sigma \\
&= \begin{cases} \mathrm{tt}, & \text{if } [\![b_1[e/x]]\!]_\sigma = \mathrm{tt} \text{ and } [\![b_2[e/x]]\!]_\sigma = \mathrm{tt} \\ \mathrm{ff}, & \text{otherwise} \end{cases} \\
&= \begin{cases} \mathrm{tt}, & \text{if } [\![b_1]\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}} = \mathrm{tt} \text{ and } [\![b_2]\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}} = \mathrm{tt} \\ \mathrm{ff}, & \text{otherwise} \end{cases} \qquad \text{(by IH1 and IH2)} \\
&= [\![b_1 \wedge b_2]\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}}.
\end{aligned}$$

– case $b_1 \vee b_2$.
Similar to the case $b_1 \wedge b_2$.

Lemma 338. *For all* σ, x, r_1, r_2, $\sigma\{x \rightsquigarrow r_1\}\{x \rightsquigarrow r_2\} = \sigma\{x \rightsquigarrow r_2\}$.

Proof. For all σ, x, r_1, r_2, we have

$$\begin{aligned}\sigma\{x \rightsquigarrow r_1\}\{x \rightsquigarrow r_2\} &= \lambda y. \begin{cases} r_2, & \text{if } y = x \\ \sigma\{x \rightsquigarrow r_1\}(y), & \text{if } y \neq x \end{cases} \\ &= \lambda y. \begin{cases} r_2, & \text{if } y = x \\ \sigma(y), & \text{if } y \neq x \end{cases} \\ &= \sigma\{x \rightsquigarrow r_2\}.\end{aligned}$$

Lemma 339. *For all σ, x, y, r_1, r_2, if $x \neq y$, then $\sigma\{x \rightsquigarrow r_1\}\{y \rightsquigarrow r_2\} = \sigma\{y \rightsquigarrow r_2\}\{x \rightsquigarrow r_1\}$.*

Proof. For all σ, x, y, r_1, r_2,

$$\begin{aligned}\sigma\{x \rightsquigarrow r_1\}\{y \rightsquigarrow r_2\} &= \lambda z. \begin{cases} r_2, & \text{if } z = y \\ \sigma\{x \rightsquigarrow r_1\}(z), & \text{if } z \neq y \end{cases} \\ &= \lambda z. \begin{cases} r_2, & \text{if } z = y \\ r_1, & \text{if } z = x \\ \sigma(z), & \text{if } z \neq y \wedge z \neq x \end{cases} \\ &= \lambda z. \begin{cases} r_1, & \text{if } z = x \\ \sigma\{y \rightsquigarrow r_2\}(z), & \text{if } z \neq x \end{cases} \\ &= \sigma\{y \rightsquigarrow r_2\}\{x \rightsquigarrow r_1\}\end{aligned}$$

Lemma 340. *For all σ, e, x, r, if $x \notin fv(e)$, then $\sigma\{x \rightsquigarrow r\}|_{fv(e)} = \sigma|_{fv(e)}$.*

Proof. For all σ, e, x, r such that $x \notin fv(e)$, to prove $\sigma\{x \rightsquigarrow r\}|_{fv(e)} = \sigma|_{fv(e)}$, we need to prove $\sigma\{x \rightsquigarrow r\}|_{fv(e)}(y) = \sigma|_{fv(e)}(y)$ for all $y \in fv(e)$. For all $y \in fv(e)$, from $x \notin fv(e)$ we know $y \neq x$, thus $\sigma\{x \rightsquigarrow r\}|_{fv(e)}(y) = \sigma\{x \rightsquigarrow r\}(y) = \sigma(y) = \sigma|_{fv(e)}(y)$.

Lemma 341. *For all $\mathbf{q}, \sigma, x, e$, $\sigma \models \mathbf{q}[e/x]$ if and only if $\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \mathbf{q}$.*

Proof. by induction on $\mathbf{q}$.

- case b.
  For all σ, x, e, by Lem. 337 we have $\sigma \models b[e/x] \iff [\![b[e/x]]\!]_\sigma = \mathrm{tt} \iff [\![b]\!]_{\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}} = \mathrm{tt} \iff \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \models b$.
- case $\neg\mathbf{q}$.
  IH: for all σ, x, e, $\mu \models \mathbf{q}[e/x]$ if and only if $\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \mathbf{q}$.
  For all σ, x, e, by IH we have $\sigma \models (\neg\mathbf{q})[e/x] \iff \sigma \models \neg\mathbf{q}[e/x] \iff \sigma \not\models \mathbf{q}[e/x] \iff \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \not\models \mathbf{q} \iff \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \neg\mathbf{q}$.
- case $\mathbf{q}_1 \wedge \mathbf{q}_2$.
  IH1: for all σ, x, e, $\sigma \models \mathbf{q}_1[e/x]$ if and only if $\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \mathbf{q}_1$.
  IH2: for all σ, x, e, $\sigma \models \mathbf{q}_2[e/x]$ if and only if $\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \mathbf{q}_2$.
  For all σ, x, e, we have

$$\begin{aligned}& \sigma \models (\mathbf{q}_1 \wedge \mathbf{q}_2)[e/x] \\ \iff\ & \sigma \models \mathbf{q}_1[e/x] \wedge \mathbf{q}_2[e/x] \\ \iff\ & (\sigma \models \mathbf{q}_1[e/x]) \wedge (\sigma \models \mathbf{q}_2[e/x]) \\ \iff\ & (\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \mathbf{q}_1) \wedge (\mu\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \mathbf{q}_2) \qquad \text{(by IH1 and IH2)} \\ \iff\ & \mu\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \mathbf{q}_1 \wedge \mathbf{q}_2.\end{aligned}$$

- case $\mathbf{q}_1 \vee \mathbf{q}_2$.
  Similar to the case $\mathbf{q}_1 \wedge \mathbf{q}_2$.
- case $\forall X.\mathbf{q}$.
  IH: for all σ, x, e, $\sigma \models \mathbf{q}[e/x]$ if and only if $\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \mathbf{q}$.
  For all σ, x, e, we need to prove $\sigma \models (\forall X.\mathbf{q})[e/x]$ if and only if $\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \forall X.\mathbf{q}$. Usually X is a logical variable, so we can assume $X \notin fv(e)$ and $X \neq x$, by Lem. 340 we know $\sigma\{X \rightsquigarrow r\}|_{fv(e)} = \sigma|_{fv(e)}$. By Lem. 252 we know $[\![e]\!]_{\sigma\{X \rightsquigarrow r\}} = [\![e]\!]_\sigma$.

$$
\begin{aligned}
\sigma \models (\forall X.\mathbf{q})[e/x] &\iff \sigma \models (\forall X.\mathbf{q})[e/x] \\
&\iff \sigma \models \forall X.\mathbf{q}[e/x] \\
&\iff \forall r.\ \sigma\{X \rightsquigarrow r\} \models \mathbf{q}[e/x] \\
&\iff \forall r.\ \sigma\{X \rightsquigarrow r\}\{x \rightsquigarrow [\![e]\!]_{\sigma\{X \rightsquigarrow r\}}\} \models \mathbf{q} \qquad \text{(by IH)} \\
&\iff \forall r.\ \sigma\{X \rightsquigarrow r\}\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \mathbf{q} \\
&\iff \forall r.\ \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}\{X \rightsquigarrow r\} \models \mathbf{q} \qquad \text{(by Lem. 339)} \\
&\iff \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \forall X.\mathbf{q}.
\end{aligned}
$$

- case $\exists X.\mathbf{q}$.
  Similar to the case $\forall X.\mathbf{q}$.

Lemma 342. *For all σ, μ, x, e, if $\sigma \in supp(\mu)$, then $\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \in supp(\mu\{x \rightsquigarrow e\})$.*

Proof. For all σ, μ, x, e such that $\sigma \in supp(\mu)$, we know $\mu(\sigma) > 0$, thus

$$
\begin{aligned}
\mu\{x \rightsquigarrow e\}(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}) &= \mathbb{E}_{\sigma_0 \sim \mu}\{\delta(\sigma_0\{x \rightsquigarrow [\![e]\!]_\sigma\})\}(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}) \\
&= \textstyle\sum_{\sigma_0} \mu(\sigma_0) \cdot \delta(\sigma_0\{x \rightsquigarrow [\![e]\!]_\sigma\})(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}) \\
&\geq \mu(\sigma) \\
&> 0,
\end{aligned}
$$

so $\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \in supp(\mu\{x \rightsquigarrow e\})$.

Lemma 343. *For all μ, x, e, σ, if $\sigma \in supp(\mu\{x \rightsquigarrow e\})$, then there exists σ_0 such that $\sigma_0 \in supp(\mu)$ and $\sigma_0\{x \rightsquigarrow [\![e]\!]_{\sigma_0}\} = \sigma$.*

Proof. For all μ, x, e, σ such that $\sigma \in supp(\mu\{x \rightsquigarrow e\})$, we know $\mu\{x \rightsquigarrow e\}(\sigma) > 0$. By Lem. 334 we know $\mu\{x \rightsquigarrow e\}(\sigma) = \sum_{\sigma_0}\{\mu(\sigma_0) \mid \sigma_0\{x \rightsquigarrow [\![e]\!]_{\sigma_0}\} = \sigma\}$, so $\sum_{\sigma_0}\{\mu(\sigma_0) \mid \sigma_0\{x \rightsquigarrow [\![e]\!]_{\sigma_0}\} = \sigma\} > 0$, thus there exists σ_0 such that $\mu(\sigma_0) > 0$ and $\sigma_0\{x \rightsquigarrow [\![e]\!]_{\sigma_0}\} = \sigma$. From $\mu(\sigma) > 0$ we know $\sigma_0 \in supp(\mu)$.

Lemma 344. *For all ξ, x, e, μ, $[\![\xi[e/x]]\!]_\mu = [\![\xi]\!]_{\mu\{x \rightsquigarrow e\}}$.*

Proof. For all ξ, x, e, μ, we prove $[\![\xi[e/x]]\!]_\mu = [\![\xi]\!]_{\mu\{x \rightsquigarrow e\}}$ by induction on ξ.

- case r.
  $[\![r[e/x]]\!]_\mu = [\![r]\!]_\mu = r = [\![r]\!]_{\mu\{x \rightsquigarrow e\}}$.

– case $\mathbb{E}(e')$.

$$\begin{aligned}
[\![\mathbb{E}(e')[e/x]]\!]_\mu &= [\![\mathbb{E}(e'[e/x])]\!]_\mu \\
&= \mathbb{E}_{\sigma\sim\mu}[[\![e'[e/x]]\!]_\sigma] \\
&= \mathbb{E}_{\sigma\sim\mu}[[\![e']\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}}] \qquad \text{(by Lem. 336)} \\
&= \textstyle\sum_\sigma \mu(\sigma)\cdot[\![e']\!]_{\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}} \\
&= \textstyle\sum_{\sigma,\sigma'}\{\mu(\sigma)\cdot[\![e']\!]_{\sigma'} \mid \sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}=\sigma'\} \\
&= \textstyle\sum_{\sigma'}(\sum_\sigma\{\mu(\sigma)\mid\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}=\sigma'\})\cdot[\![e']\!]_{\sigma'} \\
&= \textstyle\sum_{\sigma'}\mu\{x\rightsquigarrow e\}(\sigma')\cdot[\![e']\!]_{\sigma'} \qquad \text{(by Lem. 334)} \\
&= \mathbb{E}_{\sigma'\sim\mu\{x\rightsquigarrow e\}}[[\![e']\!]_{\sigma'}] \\
&= [\![\mathbb{E}(e')]\!]_{\mu\{x\rightsquigarrow e\}}.
\end{aligned}$$

– case $\mathbf{Pr}(\mathbf{q})$.

$$\begin{aligned}
[\![\mathbf{Pr}(\mathbf{q})[e/x]]\!]_\mu &= [\![\mathbf{Pr}(\mathbf{q}[e/x])]\!]_\mu \\
&= \mathbf{Pr}_{\sigma\sim\mu}[\sigma\models\mathbf{q}[e/x]] \\
&= \mathbf{Pr}_{\sigma\sim\mu}[\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}\models\mathbf{q}] \qquad \text{(by Lem. 341)} \\
&= \textstyle\sum_\sigma\{\mu(\sigma)\mid\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}\models\mathbf{q}\} \\
&= \textstyle\sum_{\sigma,\sigma'}\{\mu(\sigma)\mid\sigma'\models\mathbf{q}\wedge\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}=\sigma'\} \\
&= \textstyle\sum_{\sigma'}\{\sum_\sigma\{\mu(\sigma)\mid\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}=\sigma'\}\mid\sigma'\models\mathbf{q}\} \\
&= \textstyle\sum_{\sigma'}\{\mu\{x\rightsquigarrow e\}(\sigma')\mid\sigma'\models\mathbf{q}\} \qquad \text{(by Lem. 334)} \\
&= \mathbf{Pr}_{\sigma'\sim\mu\{x\rightsquigarrow e\}}[\sigma'\models\mathbf{q}] \\
&= [\![\mathbf{Pr}(b)]\!]_{\mu\{x\rightsquigarrow e\}}.
\end{aligned}$$

– case $\xi_1+\xi_2$.
IH1: $[\![\xi_1[e/x]]\!]_\mu = [\![\xi_1]\!]_{\mu\{x\rightsquigarrow e\}}$.
IH2: $[\![\xi_2[e/x]]\!]_\mu = [\![\xi_2]\!]_{\mu\{x\rightsquigarrow e\}}$.
By IH1 and IH2 we have $[\![(\xi_1+\xi_2)[e/x]]\!]_\mu = [\![\xi_1[e/x]]\!]_\mu + [\![\xi_1[e/x]]\!]_\mu = [\![\xi_1]\!]_{\mu\{x\rightsquigarrow e\}} + [\![\xi_2]\!]_{\mu\{x\rightsquigarrow e\}} = [\![\xi_1+\xi_2]\!]_{\mu\{x\rightsquigarrow e\}}$.
– case $\xi_1-\xi_2$.
Similar to the case $\xi_1+\xi_2$.
– case $\xi_1*\xi_2$.
Similar to the case $\xi_1+\xi_2$.

Lemma 345. *For all μ, x, e, X, r, if $X \neq x$ and $X \notin fv(e)$, then $\mu\{x\rightsquigarrow e\}\{X\rightsquigarrow r\} = \mu\{X\rightsquigarrow r\}\{x\rightsquigarrow e\}$.*

Proof. For all μ, x, e, X, r such that $X \neq x$ and $X \notin fv(e)$, we have

$$\begin{aligned}
&\mu\{x\rightsquigarrow e\}\{X\rightsquigarrow r\} \\
&= \lambda\sigma''.\textstyle\sum_{\sigma'}\{\mu\{x\rightsquigarrow e\}(\sigma')\mid\sigma'\{X\rightsquigarrow r\}=\sigma''\} \qquad \text{(by Lem. 334)} \\
&= \lambda\sigma''.\textstyle\sum_{\sigma'}\{\sum_\sigma\{\mu(\sigma)\mid\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}=\sigma'\}\mid\sigma'\{X\rightsquigarrow r\}=\sigma''\} \qquad \text{(by Lem. 334)} \\
&= \lambda\sigma''.\textstyle\sum_{\sigma,\sigma'}\{\mu(\sigma)\mid\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}=\sigma'\wedge\sigma'\{X\rightsquigarrow r\}=\sigma''\} \\
&= \lambda\sigma''.\textstyle\sum_{\sigma}\{\mu(\sigma)\mid\sigma\{x\rightsquigarrow[\![e]\!]_\sigma\}\{X\rightsquigarrow r\}=\sigma''\} \\
&= \lambda\sigma''.\textstyle\sum_{\sigma}\{\mu(\sigma)\mid\sigma\{X\rightsquigarrow r\}\{x\rightsquigarrow[\![e]\!]_\sigma\}=\sigma''\} \qquad \text{(by Lem. 339)} \\
&= \lambda\sigma''.\textstyle\sum_{\sigma,\sigma'}\{\mu(\sigma)\mid\sigma\{X\rightsquigarrow r\}=\sigma'\wedge\sigma'\{x\rightsquigarrow[\![e]\!]_\sigma\}=\sigma''\} \\
&= \lambda\sigma''.\textstyle\sum_{\sigma'}\{\sum_\sigma\{\mu(\sigma)\mid\sigma\{X\rightsquigarrow r\}=\sigma'\}\mid\sigma'\{x\rightsquigarrow[\![e]\!]_\sigma\}=\sigma''\} \\
&= \lambda\sigma''.\textstyle\sum_{\sigma'}\{\mu\{X\rightsquigarrow r\}(\sigma')\mid\sigma'\{x\rightsquigarrow e\}=\sigma''\} \qquad \text{(by Lem. 334)} \\
&= \mu\{X\rightsquigarrow r\}\{x\rightsquigarrow e\}. \qquad \text{(by Lem. 334)}
\end{aligned}$$

Lemma 346. *For all* μ_1, μ_2, p, x, e, $(\mu_1 \oplus_p \mu_2)\{x \rightsquigarrow e\} = \mu_1\{x \rightsquigarrow e\} \oplus_p \mu_2\{x \rightsquigarrow e\}$.

Proof. For all μ_1, μ_2, p, x, e, we have

$$\begin{aligned}
&(\mu_1 \oplus_p \mu_2)\{x \rightsquigarrow e\} \\
&= \mathbb{E}_{\sigma \sim \mu_1 \oplus_p \mu_2}\{\delta(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})\} \\
&= \mathbb{E}_{\sigma \sim \mu_1}\{\delta(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})\} \oplus_p \mathbb{E}_{\sigma \sim \mu_2}\{\delta(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})\} \qquad \text{(by Lem. 16)} \\
&= \mu_1\{x \rightsquigarrow e\} \oplus_p \mu_2\{x \rightsquigarrow e\}.
\end{aligned}$$

Lemma 347. *For all* $\mu, x, e, \mu_1', \mu_2', p$, *if* $\mu\{x \rightsquigarrow e\} = \mu_1' \oplus_p \mu_2'$, *then there exists* μ_1 *and* μ_2 *such that* $\mu = \mu_1 \oplus_p \mu_2$, $\mu_1\{x \rightsquigarrow e\} = \mu_1'$ *and* $\mu_2\{x \rightsquigarrow e\} = \mu_2'$.

Proof. For all $\mu, x, e, \mu_1', \mu_2', p$ such that $\mu\{x \rightsquigarrow e\} = \mu_1' \oplus_p \mu_2'$, let $\mu_1 \stackrel{\text{def}}{=} \lambda\sigma. \frac{\mu(\sigma) \cdot \mu_1'(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})}{\mu\{x \rightsquigarrow e\}(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})}$ and $\mu_2 \stackrel{\text{def}}{=} \lambda\sigma. \frac{\mu(\sigma) \cdot \mu_2'(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})}{\mu\{x \rightsquigarrow e\}(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})}$, then

$$\begin{aligned}
\mu_1 \oplus_p \mu_2 &= \lambda\sigma.\ p \cdot \mu_1(\sigma) + (1-p) \cdot \mu_2(\sigma) \\
&= \lambda\sigma.\ p \cdot \tfrac{\mu(\sigma) \cdot \mu_1'(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})}{\mu\{x \rightsquigarrow e\}(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})} + (1-p) \cdot \tfrac{\mu(\sigma) \cdot \mu_2'(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})}{\mu\{x \rightsquigarrow e\}(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})} \\
&= \lambda\sigma.\ \tfrac{\mu(\sigma)}{\mu\{x \rightsquigarrow e\}(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})} \cdot (p \cdot \mu_1'(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}) + (1-p) \cdot \mu_2'(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})) \\
&= \lambda\sigma.\ \tfrac{\mu(\sigma)}{\mu\{x \rightsquigarrow e\}(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})} \cdot (\mu_1' \oplus_p \mu_2')(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}) \\
&= \lambda\sigma.\ \tfrac{\mu(\sigma)}{\mu\{x \rightsquigarrow e\}(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})} \cdot \mu\{x \rightsquigarrow e\}(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}) \\
&= \lambda\sigma.\ \mu(\sigma) \\
&= \mu
\end{aligned}$$

and

$$\begin{aligned}
\mu_1\{x \rightsquigarrow e\} &= \lambda\sigma'. \textstyle\sum_\sigma \{\mu_1(\sigma) \mid \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} = \sigma'\} \qquad \text{(by Lem. 334)} \\
&= \lambda\sigma'. \textstyle\sum_\sigma \{\frac{\mu(\sigma) \cdot \mu_1'(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})}{\mu\{x \rightsquigarrow e\}(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})} \mid \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} = \sigma'\} \\
&= \lambda\sigma'. \textstyle\sum_\sigma \{\frac{\mu(\sigma) \cdot \mu_1'(\sigma')}{\mu\{x \rightsquigarrow e\}(\sigma')} \mid \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} = \sigma'\} \\
&= \lambda\sigma'. \tfrac{\mu_1'(\sigma')}{\mu\{x \rightsquigarrow e\}(\sigma')} \cdot \textstyle\sum_\sigma \{\mu(\sigma) \cdot \mid \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} = \sigma'\} \\
&= \lambda\sigma'. \tfrac{\mu_1'(\sigma')}{\mu\{x \rightsquigarrow e\}(\sigma')} \cdot \mu\{x \rightsquigarrow e\}(\sigma') \qquad \text{(by Lem. 334)} \\
&= \lambda\sigma'. \mu_1'(\sigma') \\
&= \mu_1'.
\end{aligned}$$

Simiarly we can prove $\mu_2\{x \rightsquigarrow e\} = \mu_2'$.

Definition 89. Let $V \in \mathbb{D}_{\mathbb{D}_{State}}$, we define $V\{x \rightsquigarrow e\} \stackrel{\text{def}}{=} \lambda\mu.\ \sum_\nu \{V(\nu) \cdot \nu\{x \rightsquigarrow e\} = \mu\}$.

Lemma 348. *For all* $V \in \mathbb{D}_{\mathbb{D}_{State}}$ *and* x, e, $\overline{V\{x \rightsquigarrow e\}} = \overline{V}\{x \rightsquigarrow e\}$.

Proof. For all $V \in \mathbb{D}_{\mathbb{D}_{State}}$ and x, e, we have

$$\begin{aligned}
\overline{V\{x \rightsquigarrow e\}} &= \lambda\sigma. \textstyle\sum_\mu V\{x \rightsquigarrow e\}(\mu) \cdot \mu(\sigma) \\
&= \lambda\sigma. \textstyle\sum_\mu \sum_\nu \{V(\nu) \cdot \mu(\sigma) \mid \nu\{x \rightsquigarrow e\} = \mu\} \\
&= \lambda\sigma. \textstyle\sum_\nu V(\nu) \cdot \nu\{x \rightsquigarrow e\}(\sigma)
\end{aligned}$$

and

$$\begin{aligned}\overline{V}\{x \rightsquigarrow e\} &= \lambda\sigma.\, \textstyle\sum_{\sigma'}\{\overline{V}(\sigma') \mid \sigma'\{x \rightsquigarrow [\![e]\!]_{\sigma'}\} = \sigma\} \qquad \text{(by Lem. 334)}\\ &= \lambda\sigma.\, \textstyle\sum_{\sigma'}\sum_{\nu}\{V(\nu)\cdot\nu(\sigma') \mid \sigma'\{x \rightsquigarrow [\![e]\!]_{\sigma'}\} = \sigma\}\\ &= \lambda\sigma.\, \textstyle\sum_{\nu} V(\nu)\cdot\sum_{\sigma'}\{\nu(\sigma') \mid \sigma'\{x \rightsquigarrow [\![e]\!]_{\sigma'}\} = \sigma\}\\ &= \lambda\sigma.\, \textstyle\sum_{\nu} V(\nu)\cdot\nu\{x \rightsquigarrow e\}(\sigma), \qquad \text{(by Lem. 334)}\end{aligned}$$

thus $\overline{V\{x \rightsquigarrow e\}} = \overline{V}\{x \rightsquigarrow e\}$.

Lemma 349. *For all $V \in \mathbb{D}_{\mathbb{D}_{State}}$ and x, e, $supp(V\{x \rightsquigarrow e\}) = \{\nu\{x \rightsquigarrow e\} \mid \nu \in supp(V)\}$.*

Proof. For all $V \in \mathbb{D}_{\mathbb{D}_{State}}$ and x, e, we have

$$\begin{aligned}supp(V\{x \rightsquigarrow e\}) &= \{\mu \mid V\{x \rightsquigarrow e\}(\mu) > 0\}\\ &= \{\mu \mid \textstyle\sum_{\nu}\{V(\nu) \mid \nu\{x \rightsquigarrow e\} = \mu\} > 0\}\\ &= \{\mu \mid \exists\nu.\, V(\nu) > 0 \wedge \nu\{x \rightsquigarrow e\} = \mu\}\\ &= \{\mu \mid \exists\nu.\, \nu \in supp(V) \wedge \nu\{x \rightsquigarrow e\} = \mu\}\\ &= \{\nu\{x \rightsquigarrow e\} \mid \nu \in supp(V)\}.\end{aligned}$$

Definition 90. $scale(\mu, \mu', x, e) \stackrel{\text{def}}{=} \lambda\sigma.\, \frac{\mu(\sigma)\cdot\mu'(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})}{\mu\{x \rightsquigarrow e\}(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})}$.

Lemma 350. *For all μ, μ', x, e, $scale(\mu, \mu', x, e)\{x \rightsquigarrow e\} = \mu'$.*

Proof. For all μ, μ', x, e, we have

$$\begin{aligned}&scale(\mu, \mu', x, e)\{x \rightsquigarrow e\}\\ &= \lambda\sigma'.\, \textstyle\sum_{\sigma}\{scale(\mu, \mu', x, e)(\sigma) \mid \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} = \sigma'\} \qquad \text{(by Lem. 334)}\\ &= \lambda\sigma'.\, \textstyle\sum_{\sigma}\{\frac{\mu(\sigma)\cdot\mu'(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})}{\mu\{x \rightsquigarrow e\}(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})} \mid \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} = \sigma'\}\\ &= \lambda\sigma'.\, \textstyle\sum_{\sigma}\{\frac{\mu(\sigma)\cdot\mu'(\sigma')}{\mu\{x \rightsquigarrow e\}(\sigma')} \mid \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} = \sigma'\}\\ &= \lambda\sigma'.\, \tfrac{\mu'(\sigma')}{\mu\{x \rightsquigarrow e\}(\sigma')} \cdot \textstyle\sum_{\sigma}\{\mu(\sigma) \mid \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} = \sigma'\}\\ &= \lambda\sigma'.\, \tfrac{\mu'(\sigma')}{\mu\{x \rightsquigarrow e\}(\sigma')} \cdot \mu\{x \rightsquigarrow e\}(\sigma') \qquad \text{(by Lem. 334)}\\ &= \lambda\sigma'.\, \mu'(\sigma')\\ &= \mu'.\end{aligned}$$

Lemma 351. *For all $V \in \mathbb{D}_{\mathbb{D}_{State}}$ and μ, x, e, if $\overline{V} = \mu\{x \rightsquigarrow e\}$, then there exists V' such that $V'\{x \rightsquigarrow e\} = V$ and $\overline{V'} = \mu$.*

Proof. For all $V \in \mathbb{D}_{\mathbb{D}_{State}}$ and μ, x, e such that $\overline{V} = \mu\{x \rightsquigarrow e\}$, let $V' \stackrel{\text{def}}{=} \lambda\nu'.\, \sum_{\nu}\{V(\nu) \mid scale(\mu, \nu) = \nu'\}$, then

$$\begin{aligned}\overline{V'} &= \lambda\sigma.\, \textstyle\sum_{\nu'} V'(\nu')\cdot\nu'(\sigma)\\ &= \lambda\sigma.\, \textstyle\sum_{\nu'}\sum_{\nu}\{V(\nu)\cdot\nu'(\sigma) \mid scale(\mu, \nu) = \nu'\}\\ &= \lambda\sigma.\, \textstyle\sum_{\nu} V(\nu)\cdot scale(\mu, \nu)(\sigma)\\ &= \lambda\sigma.\, \textstyle\sum_{\nu} V(\nu)\cdot\frac{\mu(\sigma)\cdot\nu(\sigma[\![x \rightsquigarrow [\![e]\!]_\sigma]\!])}{\mu\{x \rightsquigarrow e\}(\sigma[\![x \rightsquigarrow [\![e]\!]_\sigma]\!])}\\ &= \lambda\sigma.\, \tfrac{\mu(\sigma)}{\mu\{x \rightsquigarrow e\}(\sigma[\![x \rightsquigarrow [\![e]\!]_\sigma]\!])} \cdot \textstyle\sum_{\nu} V(\nu)\cdot\nu(\sigma[\![x \rightsquigarrow [\![e]\!]_\sigma]\!])\\ &= \lambda\sigma.\, \tfrac{\mu(\sigma)}{\mu\{x \rightsquigarrow e\}(\sigma[\![x \rightsquigarrow [\![e]\!]_\sigma]\!])} \cdot \overline{V}(\sigma[\![x \rightsquigarrow [\![e]\!]_\sigma]\!])\\ &= \lambda\sigma.\, \tfrac{\mu(\sigma)}{\mu\{x \rightsquigarrow e\}(\sigma[\![x \rightsquigarrow [\![e]\!]_\sigma]\!])} \cdot \mu\{x \rightsquigarrow e\}(\sigma[\![x \rightsquigarrow [\![e]\!]_\sigma]\!])\\ &= \lambda\sigma.\, \mu(\sigma)\\ &= \mu\end{aligned}$$

and

$$\begin{aligned} VS'\{x \rightsquigarrow e\} &= \lambda\nu''.\sum_{\nu'}\{V'(\nu') \mid \nu'\{x \rightsquigarrow e\} = \nu''\} \\ &= \lambda\nu''.\sum_{\nu'}\sum_{\nu}\{V(\nu) \mid scale(\mu,\nu) = \nu' \wedge \nu'\{x \rightsquigarrow e\} = \nu''\} \\ &= \lambda\nu''.\sum_{\nu}\{V(\nu) \mid scale(\mu,\nu)\{x \rightsquigarrow e\} = \nu''\} \\ &= \lambda\nu''.\sum_{\nu}\{V(\nu) \mid \nu = \nu''\} \qquad \text{(by Lem. 350)} \\ &= V. \end{aligned}$$

Lemma 352. *For all Q, μ, x, e, $\mu \models Q[e/x]$ if and only if $\mu\{x \rightsquigarrow e\} \models Q$.*

Proof. by induction on Q.

- case $\lceil \mathbf{q} \rceil$.
  For all μ, x, e, we have $\mu\{x \rightsquigarrow e\} \models \lceil \mathbf{q} \rceil \iff \forall \sigma \in supp(\mu\{x \rightsquigarrow e\}).\ \sigma \models \mathbf{q}$ and

  $$\begin{aligned} \mu \models \lceil \mathbf{q} \rceil[e/x] &\iff \mu \models \lceil \mathbf{q}[e/x] \rceil \\ &\iff \forall \sigma \in supp(\mu).\ \sigma \models \mathbf{q}[e/x] \\ &\iff \forall \sigma \in supp(\mu).\ \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \mathbf{q} \qquad \text{(by Lem. 341)} \end{aligned}$$

  To prove $\mu \models \lceil \mathbf{q} \rceil[e/x] \iff \mu\{x \rightsquigarrow e\} \models \mathbf{q}$, we need to prove $(\forall \sigma \in supp(\mu).\ \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \mathbf{q}) \iff (\forall \sigma \in supp(\mu\{x \rightsquigarrow e\}).\ \sigma \models \mathbf{q})$. We prove the two directions respectively.
  - $\forall \sigma \in supp(\mu).\ \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \mathbf{q}$.
    For all $\sigma \in supp(\mu\{x \rightsquigarrow e\})$, by Lem. 343 there exists σ_0 such that $\sigma_0 \in supp(\mu)$ and $\sigma_0\{x \rightsquigarrow [\![e]\!]_{\sigma_0}\} = \sigma$. From $\sigma_0 \in supp(\mu)$ we know $\sigma_0\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \mathbf{q}$, i.e., $\sigma \models \mathbf{q}$.
  - $\forall \sigma \in supp(\mu\{x \rightsquigarrow e\}).\ \sigma \models \mathbf{q}$.
    For all $\sigma \in supp(\mu)$, by Lem. 342 we have $\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \in supp(\mu\{x \rightsquigarrow e\})$, thus $\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\} \models \mathbf{q}$.
- case $\xi_1 < \xi_2$.
  For all μ, x, e, by Lem. 344 we have $\mu \models (\xi_1 < \xi_2)[e/x] \iff \mu \models \xi_1[e/x] < \xi_2[e/x] \iff [\![\xi_1[e/x]]\!]_\mu < [\![\xi_2[e/x]]\!]_\mu \iff [\![\xi_1]\!]_{\mu\{x \rightsquigarrow e\}} < [\![\xi_2]\!]_{\mu\{x \rightsquigarrow e\}} \iff \mu\{x \rightsquigarrow e\} \models \xi_1 < \xi_2$.
- case $\xi_1 = \xi_2$.
  Similar to the case $\xi_1 < \xi_2$.
- case $\xi_1 \leq \xi_2$.
  Similar to the case $\xi_1 < \xi_2$.
- case $\neg Q$.
  IH: for all μ, x, e, $\mu \models Q[e/x]$ if and only if $\mu\{x \rightsquigarrow e\} \models Q$.
  For all μ, x, e, by IH we have $\mu \models (\neg Q)[e/x] \iff \mu \models \neg Q[e/x] \iff \mu \not\models Q[e/x] \iff \mu\{x \rightsquigarrow e\} \not\models Q \iff \mu\{x \rightsquigarrow e\} \models \neg Q$.
- case $Q_1 \wedge Q_2$.
  IH1: for all μ, x, e, $\mu \models Q_1[e/x]$ if and only if $\mu\{x \rightsquigarrow e\} \models Q_1$.
  IH2: for all μ, x, e, $\mu \models Q_2[e/x]$ if and only if $\mu\{x \rightsquigarrow e\} \models Q_2$.

For all μ, x, e, we have

$$\begin{aligned}
& \mu \models (Q_1 \wedge Q_2)[e/x] \\
\iff & \mu \models Q_1[e/x] \wedge Q_2[e/x] \\
\iff & (\mu \models Q_1[e/x]) \wedge (\mu \models Q_2[e/x]) \\
\iff & (\mu\{x \rightsquigarrow e\} \models Q_1) \wedge (\mu\{x \rightsquigarrow e\} \models Q_2) \qquad \text{(by IH1 and IH2)} \\
\iff & \mu\{x \rightsquigarrow e\} \models Q_1 \wedge Q_2.
\end{aligned}$$

– case $Q_1 \vee Q_2$.
IH1: for all μ, x, e, $\mu \models Q_1[e/x]$ if and only if $\mu\{x \rightsquigarrow e\} \models Q_1$.
IH2: for all μ, x, e, $\mu \models Q_2[e/x]$ if and only if $\mu\{x \rightsquigarrow e\} \models Q_2$.
For all μ, x, e, we have

$$\begin{aligned}
& \mu \models (Q_1 \vee Q_2)[e/x] \\
\iff & \mu \models Q_1[e/x] \vee Q_2[e/x] \\
\iff & (\mu \models Q_1[e/x]) \vee (\mu \models Q_2[e/x]) \\
\iff & (\mu\{x \rightsquigarrow e\} \models Q_1) \vee (\mu\{x \rightsquigarrow e\} \models Q_2) \qquad \text{(by IH1 and IH2)} \\
\iff & \mu\{x \rightsquigarrow e\} \models Q_1 \vee Q_2.
\end{aligned}$$

– case $\forall X.Q$.
IH: for all μ, x, e, $\mu \models Q[e/x]$ if and only if $\mu\{x \rightsquigarrow e\} \models Q$.
For all μ, x, e, we need to prove $\mu \models (\forall X.Q)[e/x]$ if and only if $\mu\{x \rightsquigarrow e\} \models \forall X.Q$. Usually X is a logical variable, so we can assume $X \notin fv(e)$ and $X \neq x$, by Lem. 345 we know $\mu\{X \rightsquigarrow r\}\{x \rightsquigarrow e\} = \mu\{x \rightsquigarrow e\}\{X \rightsquigarrow r\}$, thus

$$\begin{aligned}
\mu \models (\forall X.Q)[e/x] & \iff \mu \models (\forall X.Q)[e/x] \\
& \iff \mu \models \forall X.Q[e/x] \\
& \iff \forall r.\ \mu\{X \rightsquigarrow r\} \models Q[e/x] \\
& \iff \forall r.\ \mu\{X \rightsquigarrow r\}\{x \rightsquigarrow e\} \models Q \qquad \text{(by IH)} \\
& \iff \forall r.\ \mu\{x \rightsquigarrow e\}\{X \rightsquigarrow r\} \models Q \\
& \iff \mu\{x \rightsquigarrow e\} \models \forall X.Q.
\end{aligned}$$

– case $\exists X.Q$.
Similar to the case $\forall X.Q$.
– case $Q_1 \oplus_p Q_2$.
IH1: for all μ, x, e, $\mu \models Q_1[e/x]$ if and only if $\mu\{x \rightsquigarrow e\} \models Q_1$.
IH2: for all μ, x, e, $\mu \models Q_2[e/x]$ if and only if $\mu\{x \rightsquigarrow e\} \models Q_2$.
For all μ, x, e, we need to prove $\mu \models (Q_1 \oplus_p Q_2)[e/x]$ if and only if $\mu\{x \rightsquigarrow e\} \models (Q_1 \oplus_p Q_2)$. There are three cases: $p = 0$, $p = 1$ and $0 < p < 1$. We prove the three cases respectively.

- case $p = 0$.

$$\begin{aligned}
& \mu \models (Q_1 \oplus_0 Q_2)[e/x] \\
\iff & \mu \models Q_1[e/x] \oplus_0 Q_2[e/x] \\
\iff & (\mu \models Q_2[e/x]) \\
\iff & (\mu\{x \rightsquigarrow e\} \models Q_2) \qquad \text{(by IH2)} \\
\iff & \mu\{x \rightsquigarrow e\} \models Q_1 \oplus_0 Q_2.
\end{aligned}$$

- case $p = 1$.

$$\begin{aligned}
& \mu \models (Q_1 \oplus_1 Q_2)[e/x] \\
\iff & \mu \models Q_1[e/x] \oplus_1 Q_2[e/x] \\
\iff & (\mu \models Q_1[e/x]) \\
\iff & (\mu\{x \rightsquigarrow e\} \models Q_1) \qquad \text{(by IH1)} \\
\iff & \mu\{x \rightsquigarrow e\} \models Q_1 \oplus_1 Q_2.
\end{aligned}$$

- case $0 < p < 1$.

$$\begin{aligned}
& \mu \models (Q_1 \oplus_p Q_2)[e/x] \\
\iff & \mu \models Q_1[e/x] \oplus_p Q_2[e/x] \\
\iff & \exists \mu_1, \mu_2.\ \mu = \mu_1 \oplus_p \mu_2 \wedge (\mu_1 \models Q_1[e/x]) \wedge (\mu_2 \models Q_2[e/x]) \\
\iff & \exists \mu_1, \mu_2.\ \mu = \mu_1 \oplus_p \mu_2 \wedge (\mu_1\{x \rightsquigarrow e\} \models Q_1) \wedge (\mu_2\{x \rightsquigarrow e\} \models Q_2) \qquad \text{(by IH1 and IH2)} \\
\iff & \exists \mu_1, \mu_2.\ \mu = \mu_1 \oplus_p \mu_2 \wedge \mu\{x \rightsquigarrow e\} = \mu_1\{x \rightsquigarrow e\} \oplus_p \mu_2\{x \rightsquigarrow e\} \wedge \\
& \qquad (\mu_1\{x \rightsquigarrow e\} \models Q_1) \wedge (\mu_2\{x \rightsquigarrow e\} \models Q_2) \qquad \text{(by Lem. 346)} \\
\iff & \exists \mu_1', \mu_2', \mu_1, \mu_2.\ \mu = \mu_1 \oplus_p \mu_2 \wedge \mu_1\{x \rightsquigarrow e\} = \mu_1' \wedge \mu_2\{x \rightsquigarrow e\} = \mu_2' \wedge \\
& \qquad \mu\{x \rightsquigarrow e\} = \mu_1' \oplus_p \mu_2' \wedge (\mu_1' \models Q_1) \wedge (\mu_2' \models Q_2) \\
\iff & \exists \mu_1', \mu_2'.\ \mu\{x \rightsquigarrow e\} = \mu_1' \oplus_p \mu_2' \wedge (\mu_1' \models Q_1) \wedge (\mu_2' \models Q_2) \qquad \text{(by Lem. 347)} \\
\iff & \mu\{x \rightsquigarrow e\} \models Q_1 \oplus_p Q_2.
\end{aligned}$$

– case $Q_1 \oplus Q_2$.
IH1: for all μ, x, e, $\mu \models Q_1[e/x]$ if and only if $\mu\{x \rightsquigarrow e\} \models Q_1$.
IH2: for all μ, x, e, $\mu \models Q_2[e/x]$ if and only if $\mu\{x \rightsquigarrow e\} \models Q_2$.
For all μ, x, e, we have $\mu\{x \rightsquigarrow e\} \models Q_1 \oplus Q_2 \iff \exists p.\ \mu\{x \rightsquigarrow e\} \models Q_1 \oplus_p Q_2$ and

$$\begin{aligned}
\mu \models (Q_1 \oplus Q_2)[e/x] & \iff \mu \models Q_1[e/x] \oplus Q_2[e/x] \\
& \iff \exists p.\ \mu \models Q_1[e/x] \oplus_p Q_2[e/x] \\
& \iff \exists p.\ \mu \models (Q_1 \oplus_p Q_2)[e/x].
\end{aligned}$$

To prove $\mu \models (Q_1 \oplus Q_2)[e/x] \iff \mu\{x \rightsquigarrow e\} \models Q_1 \oplus Q_2$, it suffices to prove $\mu \models (Q_1 \oplus_p Q_2)[e/x] \iff \mu\{x \rightsquigarrow e\} \models Q_1 \oplus_p Q_2$ for all p. The rest of the proof is similar to the case $Q_1 \oplus_p Q_2$.

– case $\bigoplus Q$.
IH1: for all μ, x, e, $\mu \models Q[e/x]$ if and only if $\mu\{x \rightsquigarrow e\} \models Q$.
For all μ, x, e, we have

$$\begin{aligned}
& \mu \models (\textstyle\bigoplus Q)[e/x] \\
\iff & \mu \models \textstyle\bigoplus Q[e/x] \\
\iff & \exists V \in \mathbb{D}_{\mathbb{D}_{State}}.\ \mu = \overline{V} \wedge (\forall \nu \in \mathit{supp}(V).\ \nu \models Q[e/x]) \\
\iff & \exists V \in \mathbb{D}_{\mathbb{D}_{State}}.\ \mu = \overline{V} \wedge (\forall \nu \in \mathit{supp}(V).\ \nu\{x \rightsquigarrow e\} \models Q) \qquad \text{(by IH)} \\
\iff & \exists V \in \mathbb{D}_{\mathbb{D}_{State}}.\ \mu = \overline{V} \wedge \mu\{x \rightsquigarrow e\} = \overline{V}\{x \rightsquigarrow e\} \wedge (\forall \nu \in \mathit{supp}(V).\ \nu\{x \rightsquigarrow e\} \models Q) \\
\iff & \exists V \in \mathbb{D}_{\mathbb{D}_{State}}.\ \mu = \overline{V} \wedge \mu\{x \rightsquigarrow e\} = \overline{V\{x \rightsquigarrow e\}} \wedge \\
& \qquad (\forall \nu \in \mathit{supp}(V\{x \rightsquigarrow e\}).\ \nu \models Q) \qquad \text{(by Lem. 348 and Lem. 349)} \\
\iff & \exists V, V' \in \mathbb{D}_{\mathbb{D}_{State}}.\ \mu = \overline{V} \wedge V\{x \rightsquigarrow e\} = V' \wedge \mu\{x \rightsquigarrow e\} = \overline{V'} \wedge (\forall \nu \in \mathit{supp}(V').\ \nu \models Q) \\
\iff & \exists V' \in \mathbb{D}_{\mathbb{D}_{State}}.\ \mu\{x \rightsquigarrow e\} = \overline{V'} \wedge (\forall \nu \in \mathit{supp}(V').\ \nu \models Q) \qquad \text{(by Lem. 351)} \\
\iff & \mu\{x \rightsquigarrow e\} \models \textstyle\bigoplus Q.
\end{aligned}$$

Lemma 353 (Soundness of (SQ-ASGN) rule). *For all x, e, P, Q, G, if $P \Rightarrow Q[e/x]$ and $(\sigma, \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}) \models G$ for all σ and μ such that $\sigma \in \mathit{supp}(\mu)$ and $\mu \models P$, then $G \models_{SQ} \{P\}x := e\{Q\}$.*

Proof. For all x, e, P, Q, G such that $P \Rightarrow Q[e/x]$ and $(\sigma, \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}) \models G$ for all σ and μ such that $\sigma \in \mathit{supp}(\mu)$ and $\mu \models P$, to prove $G \models_{\mathrm{sQ}} \{P\}x := e\{Q\}$, we need to prove for all μ, if $\mu \models P$ and $|[\![x := e]\!](\mu)| = 1$, then $[\![x := e]\!](\mu) \models Q$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![x := e]\!](\sigma))$. For all μ such that $\mu \models P$ and $|[\![x := e]\!](\mu)| = 1$, by Lem. 335 we know $[\![x := e]\!](\mu) = \mu\{x \rightsquigarrow e\}$. From $\mu \models P$ and $P \Rightarrow Q[e/x]$ we know $\mu \models Q[e/x]$. By Lem. 352 we know $\mu\{x \rightsquigarrow e\} \models Q$, thus $[\![x := e]\!](\mu) \models Q$. For all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![x := e]\!](\sigma))$, by Lem. 323 we know $[\![x := e]\!](\sigma) = \delta(\sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})$. From $\sigma' \in \mathit{supp}([\![x := e]\!](\sigma))$ we know $\sigma' = \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}$. From $\sigma \in \mathit{supp}(\mu)$ and $\mu \models P$ we know $(\sigma, \sigma') \models G$.

Lemma 354. *For all b, C_1, C_2, σ, n, if $\sigma \models b$, then $(\textbf{\textit{if}}\,(b)\ \textbf{\textit{then}}\,C_1\ \textbf{\textit{else}}\,C_2, \sigma) \xrightarrow{p}{}^{n+1}(\textbf{\textit{skip}}, \sigma')$ if and only if $(C_1, \sigma) \xrightarrow{p}{}^{n}(\textbf{\textit{skip}}, \sigma')$.*

Proof. For all b, C_1, C_2, σ, n such that $\sigma \models b$, we know $[\![b]\!]_\sigma = \mathrm{tt}$, thus

$$
\begin{aligned}
& (\textbf{if}\,(b)\ \textbf{then}\,C_1\ \textbf{else}\,C_2, \sigma) \xrightarrow{p}{}^{n+1}(\textbf{skip}, \sigma') \\
\iff\ & p = \textstyle\sum_{C', \sigma'}\{p_1 \cdot p_2 \mid (\textbf{if}\,(b)\ \textbf{then}\,C_1\ \textbf{else}\,C_2, \sigma) \xrightarrow{p_1} (C', \sigma') \wedge (C', \sigma') \xrightarrow{p_2}{}^{n}(\textbf{skip}, \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})\} \\
\iff\ & p = \textstyle\sum\{p_2 \mid (C_1, \sigma) \xrightarrow{p_2}{}^{n}(\textbf{skip}, \sigma')\} \\
\iff\ & (C_1, \sigma) \xrightarrow{p}{}^{n}(\textbf{skip}, \sigma').
\end{aligned}
$$

Lemma 355. *For all b, C_1, C_2, σ, n, if $\sigma \models \neg b$, then $(\textbf{\textit{if}}\,(b)\ \textbf{\textit{then}}\,C_1\ \textbf{\textit{else}}\,C_2, \sigma) \xrightarrow{p}{}^{n+1}(\textbf{\textit{skip}}, \sigma')$ if and only if $(C_2, \sigma) \xrightarrow{p}{}^{n}(\textbf{\textit{skip}}, \sigma')$.*

Proof. For all b, C_1, C_2, σ, n such that $\sigma \models \neg b$, we know $[\![b]\!]_\sigma = \mathrm{ff}$, thus

$$
\begin{aligned}
& (\textbf{if}\,(b)\ \textbf{then}\,C_1\ \textbf{else}\,C_2, \sigma) \xrightarrow{p}{}^{n+1}(\textbf{skip}, \sigma') \\
\iff\ & p = \textstyle\sum_{C', \sigma'}\{p_1 \cdot p_2 \mid (\textbf{if}\,(b)\ \textbf{then}\,C_1\ \textbf{else}\,C_2, \sigma) \xrightarrow{p_1} (C', \sigma') \wedge (C', \sigma') \xrightarrow{p_2}{}^{n}(\textbf{skip}, \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})\} \\
\iff\ & p = \textstyle\sum\{p_2 \mid (C_2, \sigma) \xrightarrow{p_2}{}^{n}(\textbf{skip}, \sigma')\} \\
\iff\ & (C_2, \sigma) \xrightarrow{p}{}^{n}(\textbf{skip}, \sigma').
\end{aligned}
$$

Lemma 356. *For all b, C_1, C_2, σ, if $\sigma \models b$, then $[\![\textbf{\textit{if}}\,(b)\ \textbf{\textit{then}}\,C_1\ \textbf{\textit{else}}\,C_2]\!](\sigma) = [\![C_1]\!](\sigma)$.*

Proof. For all b, C_1, C_2, σ such that $\sigma \models b$, we have

$$
\begin{aligned}
& [\![\textbf{if}\,(b)\ \textbf{then}\,C_1\ \textbf{else}\,C_2]\!](\sigma) \\
=\ & \lambda\sigma'.\lim \vec{p}, \text{where } \forall n.\ (\textbf{if}\,(b)\ \textbf{then}\,C_1\ \textbf{else}\,C_2, \sigma) \xrightarrow{\vec{p}\,[n]}{}^{n}(\textbf{skip}, \sigma') \\
=\ & \lambda\sigma'.\lim\,(0 :: \vec{p}), \text{where } \forall n.\ (\textbf{if}\,(b)\ \textbf{then}\,C_1\ \textbf{else}\,C_2, \sigma) \xrightarrow{\vec{p}\,[n]}{}^{n+1}(\textbf{skip}, \sigma') \\
=\ & \lambda\sigma'.\lim\,(0 :: \vec{p}), \text{where } \forall n.\ (C_1, \sigma) \xrightarrow{\vec{p}\,[n]}{}^{n}(\textbf{skip}, \sigma') \qquad \text{(by Lem. 354)} \\
=\ & \lambda\sigma'.\lim\ \vec{p}, \text{where } \forall n.\ (C_1, \sigma) \xrightarrow{\vec{p}\,[n]}{}^{n}(\textbf{skip}, \sigma') \\
=\ & [\![C_1]\!](\sigma).
\end{aligned}
$$

Lemma 357. *For all b, C_1, C_2, μ, if $\mu \models \lceil b \rceil$, then $[\![\textbf{\textit{if}}\,(b)\ \textbf{\textit{then}}\,C_1\ \textbf{\textit{else}}\,C_2]\!](\mu) = [\![C_1]\!](\mu)$.*

Proof. For all b, C_1, C_2, μ such that $\mu \models \lceil b \rceil$, we know $\sigma \models b$ for all $\sigma \in \mathit{supp}(\mu)$, thus

$$
\begin{aligned}
&[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu)\\
&= \mathbb{E}_{\sigma\sim\mu}\{[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma)\}\\
&= \lambda\sigma'.\textstyle\sum_\sigma \mu(\sigma)\cdot[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma)(\sigma')\\
&= \lambda\sigma'.\textstyle\sum_\sigma \{\mu(\sigma)\cdot[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma)(\sigma') \mid \sigma \in \mathit{supp}(\mu)\}\\
&= \lambda\sigma'.\textstyle\sum_\sigma \{\mu(\sigma)\cdot[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma)(\sigma') \mid \sigma \in \mathit{supp}(\mu) \wedge \sigma \models b\}\\
&= \lambda\sigma'.\textstyle\sum_\sigma \{\mu(\sigma)\cdot[\![C_1]\!](\sigma)(\sigma') \mid \sigma \in \mathit{supp}(\mu) \wedge \sigma \models b\} \qquad \text{(by Lem. 356)}\\
&= \lambda\sigma'.\textstyle\sum_\sigma \{\mu(\sigma)\cdot[\![C_1]\!](\sigma)(\sigma') \mid \sigma \in \mathit{supp}(\mu)\}\\
&= \lambda\sigma'.\textstyle\sum_\sigma \mu(\sigma)\cdot[\![C_1]\!](\sigma)(\sigma')\\
&= \mathbb{E}_{\sigma\sim\mu}\{[\![C_1]\!](\sigma)\}\\
&= [\![C_1]\!](\mu).
\end{aligned}
$$

Lemma 358. *For all b, C_1, C_2, σ, if $\sigma \models \neg b$, then $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma) = [\![C_2]\!](\sigma)$.*

Proof. For all b, C_1, C_2, σ such that $\sigma \models \neg b$, we have

$$
\begin{aligned}
&[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma)\\
&= \lambda\sigma'.\lim \vec{p}, \text{where } \forall n.\ (\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2, \sigma) \xrightarrow{\vec{p}\,[n]}{}^{n} (\textbf{skip}, \sigma')\\
&= \lambda\sigma'.\lim (0 :: \vec{p}), \text{where } \forall n.\ (\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2, \sigma) \xrightarrow{\vec{p}\,[n]}{}^{n+1} (\textbf{skip}, \sigma')\\
&= \lambda\sigma'.\lim (0 :: \vec{p}), \text{where } \forall n.\ (C_1, \sigma) \xrightarrow{\vec{p}\,[n]}{}^{n} (\textbf{skip}, \sigma') \qquad \text{(by Lem. 355)}\\
&= \lambda\sigma'.\lim\ \vec{p}, \text{where } \forall n.\ (C_1, \sigma) \xrightarrow{\vec{p}\,[n]}{}^{n} (\textbf{skip}, \sigma')\\
&= [\![C_1]\!](\sigma).
\end{aligned}
$$

Lemma 359. *For all b, C_1, C_2, μ, if $\mu \models \lceil b \rceil$, then $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu) = [\![C_2]\!](\mu)$.*

Proof. For all b, C_1, C_2, μ such that $\mu \models \lceil \neg b \rceil$, we know $\sigma \models \neg b$ for all $\sigma \in \mathit{supp}(\mu)$, thus

$$
\begin{aligned}
&[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu)\\
&= \mathbb{E}_{\sigma\sim\mu}\{[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma)\}\\
&= \lambda\sigma'.\textstyle\sum_\sigma \mu(\sigma)\cdot[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma)(\sigma')\\
&= \lambda\sigma'.\textstyle\sum_\sigma \{\mu(\sigma)\cdot[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma)(\sigma') \mid \sigma \in \mathit{supp}(\mu)\}\\
&= \lambda\sigma'.\textstyle\sum_\sigma \{\mu(\sigma)\cdot[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma)(\sigma') \mid \sigma \in \mathit{supp}(\mu) \wedge \sigma \models \neg b\}\\
&= \lambda\sigma'.\textstyle\sum_\sigma \{\mu(\sigma)\cdot[\![C_2]\!](\sigma)(\sigma') \mid \sigma \in \mathit{supp}(\mu) \wedge \sigma \models b\} \qquad \text{(by Lem. 358)}\\
&= \lambda\sigma'.\textstyle\sum_\sigma \{\mu(\sigma)\cdot[\![C_2]\!](\sigma)(\sigma') \mid \sigma \in \mathit{supp}(\mu)\}\\
&= \lambda\sigma'.\textstyle\sum_\sigma \mu(\sigma)\cdot[\![C_2]\!](\sigma)(\sigma')\\
&= \mathbb{E}_{\sigma\sim\mu}\{[\![C_2]\!](\sigma)\}\\
&= [\![C_2]\!](\mu).
\end{aligned}
$$

Lemma 360 (Soundness of (SQ-COND) rule). *For all $b, C_1, C_2, P_1, P_2, Q_1, Q_2, G$, if $G \models_{SQ} \{P_1 \wedge \lceil b \rceil\} C_1 \{Q_1\}$ and $G \models_{SQ} \{P_2 \wedge \lceil b \rceil\} C_1 \{Q_2\}$, then $G \models_{SQ} \{(P_1 \wedge \lceil b \rceil) \oplus_p (P_2 \wedge \lceil \neg b \rceil)\}$ **if** (b) **then** C_1 **else** $C_2 \{Q_1 \oplus_p Q_2\}$.*

Proof. For all $b, C_1, C_2, P_1, P_2, Q_1, Q_2, G$ such that $G \models_{\text{SQ}} \{P_1 \wedge \lceil b \rceil\} C_1 \{Q_1\}$ and $G \models_{\text{SQ}} \{P_2 \wedge \lceil b \rceil\} C_1 \{Q_2\}$, to prove $G \models_{\text{SQ}} \{(P_1 \wedge \lceil b \rceil) \oplus_p (P_2 \wedge \lceil \neg b \rceil)\}$**if** (b) **then** C_1 **else** $C_2 \{Q_1 \oplus_p Q_2\}$, we need to prove for all μ, if $\mu \models (P_1 \wedge \lceil b \rceil) \oplus_p (P_2 \wedge \lceil \neg b \rceil)$ and $|[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu)| = 1$, then $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu) \models Q_1 \oplus_p Q_2$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma))$. For all μ such that $\mu \models (P_1 \wedge \lceil b \rceil) \oplus_p (P_2 \wedge \lceil \neg b \rceil)$ and $|[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu)| = 1$, there are three cases.

- $p = 1$ and $\mu \models P_1 \wedge \lceil b \rceil$.
  From $\mu \models \lceil b \rceil$ by Lem. 357 we know $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu) = [\![C_1]\!](\mu)$. From $|[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu)| = 1$ we know $|[\![C_1]\!](\mu)| = 1$. From $G \models_{\text{SQ}} \{P_1 \wedge \lceil b \rceil\} C_1 \{Q_1\}$ and $\mu \models P_1 \wedge \lceil b \rceil$ we know $[\![C_1]\!](\mu) \models Q_1$. From $p = 1$ we have $[\![C_1]\!](\mu) \models Q_1 \oplus_p Q_2$, thus $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu) \models Q_1 \oplus_p Q_2$. For all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma))$, from $\mu \models \lceil b \rceil$ and $\sigma \in \mathit{supp}(\mu)$ we know $\sigma \models b$. By Lem. 356 we know $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma) = [\![C_1]\!](\sigma)$.
  From $\sigma' \in \mathit{supp}([\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma))$ we know $\sigma' \in \mathit{supp}([\![C_1]\!](\sigma))$. From $G \models_{\text{SQ}} \{P_1 \wedge \lceil b \rceil\} C_1 \{Q_1\}$, $\mu \models P_1$, $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C_1]\!](\sigma))$ we know $(\sigma, \sigma') \models G$.
- $p = 0$ and $\mu \models P_2 \wedge \lceil \neg b \rceil$.
  From $\mu \models \lceil \neg b \rceil$ by Lem. 359 we know $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu) = [\![C_2]\!](\mu)$. From $|[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu)| = 1$ we know $|[\![C_2]\!](\mu)| = 1$. From $G \models_{\text{SQ}} \{P_2 \wedge \lceil b \rceil\} C_2 \{Q_2\}$ and $\mu \models P_2 \wedge \lceil \neg b \rceil$ we know $[\![C_2]\!](\mu) \models Q_2$. From $p = 0$ we have $[\![C_2]\!](\mu) \models Q_1 \oplus_p Q_2$, thus $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu) \models Q_1 \oplus_p Q_2$. For all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma))$, from $\mu \models \lceil \neg b \rceil$ and $\sigma \in \mathit{supp}(\mu)$ we know $\sigma \models \neg b$. By Lem. 358 we know $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma) = [\![C_2]\!](\sigma)$. From $\sigma' \in \mathit{supp}([\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma))$ we know $\sigma' \in \mathit{supp}([\![C_2]\!](\sigma))$. From $G \models_{\text{SQ}} \{P_2 \wedge \lceil \neg b \rceil\} C_2 \{Q_2\}$, $\mu \models P_2$, $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C_2]\!](\sigma))$ we know $(\sigma, \sigma') \models G$.
- $0 < p < 1$ and there exists μ_1 and μ_2 such that $\mu = \mu_1 \oplus_p \mu_2$, $\mu_1 \models P_1 \wedge \lceil b \rceil$ and $\mu_2 \models P_2 \wedge \lceil \neg b \rceil$.
  By Lem. 313 we know $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu) = [\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu_1 \oplus_p \mu_2) = [\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu_1) \oplus_p [\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu_2)$. From $0 < p < 1$ by Lem. 314 we know $|[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu_1)| = 1$ and $|[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu_2)| = 1$. From $G \models_{\text{SQ}} \{P_1 \wedge \lceil b \rceil\} C_1 \{Q_1\}$, $\mu_1 \models P_1 \wedge \lceil b \rceil$ and $|[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu_1)| = 1$ we know $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu_1) \models Q_1$. From $G \models_{\text{SQ}} \{P_2 \wedge \lceil b \rceil\} C_1 \{Q_2\}$, $\mu_2 \models P_2 \wedge \lceil b \rceil$ and $|[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu_2)| = 1$ we know $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu_2) \models Q_2$. From $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu) = [\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu_1) \oplus_p [\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu_2)$, $0 < p < 1$, $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu_1) \models Q_1$ and $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu_2) \models Q_2$ we know $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\mu) \models Q_1 \oplus_p Q_2$. For all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma))$, from $0 < p < 1$ by Lem. 275 we know $\mathit{supp}(\mu) = \mathit{supp}(\mu_1 \oplus_p \mu_2) = \mathit{supp}(\mu_1) \cup \mathit{supp}(\mu_2)$. From $\sigma \in \mathit{supp}(\mu)$ we know $\sigma \in \mathit{supp}(\mu_1)$ or $\sigma \in \mathit{supp}(\mu_2)$. If $\sigma \in \mathit{supp}(\mu_1)$, from $\mu_1 \models \lceil b \rceil$ we know $\sigma \models b$. By Lem. 356 we know $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma) = [\![C_1]\!](\sigma)$. From $\sigma' \in \mathit{supp}([\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma))$ we know $\sigma' \in \mathit{supp}([\![C_1]\!](\sigma))$.

From $G \models_{\text{SQ}} \{P_1 \wedge \lceil b \rceil\} C_1 \{Q_1\}$, $\mu_1 \models P_1$, $\sigma \in \mathit{supp}(\mu_1)$ and $\sigma' \in \mathit{supp}([\![C_1]\!](\sigma))$ we know $(\sigma, \sigma') \models G$. If $\sigma \in \mathit{supp}(\mu_2)$, from $\mu_2 \models \lceil \neg b \rceil$ we know $\sigma \models b$. By Lem. 356 we know $[\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma) = [\![C_2]\!](\sigma)$. From $\sigma' \in \mathit{supp}([\![\textbf{if } (b) \textbf{ then } C_1 \textbf{ else } C_2]\!](\sigma))$ we know $\sigma' \in \mathit{supp}([\![C_2]\!](\sigma))$. From $G \models_{\text{SQ}} \{P_2 \wedge \lceil \neg b \rceil\} C_2 \{Q_2\}$, $\mu_2 \models P_2$, $\sigma \in \mathit{supp}(\mu_2)$ and $\sigma' \in \mathit{supp}([\![C_2]\!](\sigma))$ we know $(\sigma, \sigma') \models G$.

Lemma 361. *For all C and μ, $[\![\langle C \rangle]\!](\mu) = [\![C]\!](\mu)$.*

Proof. For all C and μ, we have $[\![\langle C \rangle]\!](\mu) = \mathbb{E}_{\sigma \sim \mu}\{[\![\langle C \rangle]\!](\sigma)\} = \mathbb{E}_{\sigma \sim \mu}\{[\![C]\!](\sigma)\} = [\![C]\!](\mu)$.

Lemma 362 (Soundness of (SQ-ATOM) rule). *For all C, P, Q, G, if $G \models_{\text{SQ}} \{P\} C \{Q\}$, then $G \models_{\text{SQ}} \{P\} \langle C \rangle \{Q\}$.*

Proof. For all C, P, Q, G such that $G \models_{\text{SQ}} \{P\} C \{Q\}$, to prove $G \models_{\text{SQ}} \{P\} \langle C \rangle \{Q\}$, we need to prove for all μ, if $\mu \models P$ and $|[\![\langle C \rangle]\!](\mu)| = 1$, then $[\![\langle C \rangle]\!](\mu) \models Q$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![\langle C \rangle]\!](\sigma))$. For all μ such that $\mu \models P$ and $|[\![\langle C \rangle]\!](\mu)| = 1$, by Lem. 361 we know $[\![\langle C \rangle]\!](\mu) = [\![C]\!](\mu)$. From $|[\![\langle C \rangle]\!](\mu)| = 1$ we know $|[\![C]\!](\mu)| = 1$. From $G \models_{\text{SQ}} \{P\} C \{Q\}$, $\mu \models P$ and $|[\![C]\!](\mu)| = 1$ we know $[\![C]\!](\mu) \models Q$, thus $[\![\langle C \rangle]\!](\mu) \models Q$. For all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![\langle C \rangle]\!](\sigma))$, we have $[\![\langle C \rangle]\!](\sigma) = [\![C]\!](\sigma)$. From $\sigma' \in \mathit{supp}([\![\langle C \rangle]\!](\sigma))$ we know $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$. From $G \models_{\text{SQ}} \{P\} C \{Q\}$, $\mu \models P$, $|[\![C]\!](\mu)| = 1$, $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![C]\!](\sigma))$ we know $(\sigma, \sigma') \models G$.

Lemma 363. *For all $C_1, C_2, p, \sigma, n, p'$, $(\langle C_1 \rangle \oplus_p \langle C_2 \rangle, \sigma) \xrightarrow{p'}{}^{n+1} (\boldsymbol{skip}, \sigma')$ if and only if there exists p_1 and p_2 such that $p' = p \cdot p_1 + (1-p) \cdot p_2$, $(\langle C_1 \rangle, \sigma) \xrightarrow{p_1}{}^{n} (\boldsymbol{skip}, \sigma')$ and $(\langle C_2 \rangle, \sigma) \xrightarrow{p_2}{}^{n} (\boldsymbol{skip}, \sigma')$.*

Proof. For all $C_1, C_2, p, \sigma, n, p'$, we have

$$
\begin{aligned}
& (\langle C_1 \rangle \oplus_p \langle C_2 \rangle, \sigma) \xrightarrow{p'}{}^{n+1} (\mathbf{skip}, \sigma') \\
\iff\ & p' = \textstyle\sum_{C', \sigma'} \{p_1 \cdot p_2 \mid (\langle C_1 \rangle \oplus_p \langle C_2 \rangle, \sigma) \xrightarrow{p_1} (C', \sigma') \wedge (C', \sigma') \xrightarrow{p_2}{}^{n} (\mathbf{skip}, \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\})\} \\
\iff\ & p' = p \cdot \textstyle\sum \{p_1 \mid (\langle C_1 \rangle, \sigma) \xrightarrow{p_1}{}^{n} (\mathbf{skip}, \sigma')\} + (1-p) \cdot \sum \{p_2 \mid (\langle C_2 \rangle, \sigma) \xrightarrow{p_2}{}^{n} (\mathbf{skip}, \sigma')\} \\
\iff\ & \exists p_1, p_2.\ p' = p \cdot p_1 + (1-p) \cdot p_2 \wedge (\langle C_1 \rangle, \sigma) \xrightarrow{p_1}{}^{n} (\mathbf{skip}, \sigma') \wedge (\langle C_2 \rangle, \sigma) \xrightarrow{p_2}{}^{n} (\mathbf{skip}, \sigma').
\end{aligned}
$$

Lemma 364. *For all C_1, C_2, σ, $[\![\langle C_1 \rangle \oplus_p \langle C_2 \rangle]\!](\sigma) = [\![\langle C_1 \rangle]\!](\sigma) \oplus_p [\![\langle C_2 \rangle]\!](\sigma)$.*

Proof. For all C_1, C_2, σ, n, we have

$$
\begin{aligned}
&[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\sigma)\\
&= \lambda\sigma'.\, \lim \vec{p}, \text{where } \forall n.\ (\langle C_1\rangle \oplus_p \langle C_2\rangle, \sigma) \xrightarrow{\vec{p}[n]}{}^{n} (\mathbf{skip}, \sigma')\\
&= \lambda\sigma'.\, \lim\ (0 :: \vec{p}), \text{where } \forall n.\ (\langle C_1\rangle \oplus_p \langle C_2\rangle, \sigma) \xrightarrow{\vec{p}[n]}{}^{n+1} (\mathbf{skip}, \sigma')\\
&= \lambda\sigma'.\, \lim \vec{p}, \text{where } \forall n.\ (\langle C_1\rangle \oplus_p \langle C_2\rangle, \sigma) \xrightarrow{\vec{p}[n]}{}^{n+1} (\mathbf{skip}, \sigma')\\
&= \lambda\sigma'.\, \lim\ (p \cdot \vec{p_1} + (1-p)\cdot \vec{p_2}), \text{where } \forall n.\ (\langle C_1\rangle, \sigma) \xrightarrow{\vec{p_1}[n]}{}^{n} (\mathbf{skip}, \sigma') \wedge\\
&\qquad (\langle C_2\rangle, \sigma) \xrightarrow{\vec{p_2}[n]}{}^{n} (\mathbf{skip}, \sigma') \qquad \text{(by Lem. 363)}\\
&= \lambda\sigma'.\, p \cdot \lim \vec{p_1} + (1-p)\cdot \lim \vec{p_2}, \text{where } \forall n.\ (\langle C_1\rangle, \sigma) \xrightarrow{\vec{p_1}[n]}{}^{n} (\mathbf{skip}, \sigma') \wedge\\
&\qquad (\langle C_2\rangle, \sigma) \xrightarrow{\vec{p_2}[n]}{}^{n} (\mathbf{skip}, \sigma')\\
&= \lambda\sigma'.\, p \cdot \lim \vec{p_1}, \text{where } \forall n.\ (C_1, \sigma) \xrightarrow{\vec{p_1}[n]}{}^{n} (\mathbf{skip}, \sigma') +\\
&\qquad (1-p)\cdot \lim \vec{p_2}, \text{where } \forall n.\ (C_1, \sigma) \xrightarrow{\vec{p_1}[n]}{}^{n} (\mathbf{skip}, \sigma')\\
&= \lambda\sigma'.\, p \cdot [\![C_1]\!](\sigma)(\sigma') + (1-p)\cdot [\![C_2]\!](\sigma)(\sigma')\\
&= [\![C_1]\!](\sigma) \oplus_p [\![C_2]\!](\sigma).
\end{aligned}
$$

Lemma 365. *For all C_1, C_2, μ, $[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu) = [\![\langle C_1\rangle]\!](\mu) \oplus_p [\![\langle C_2\rangle]\!](\mu)$.*

Proof. For all C_1, C_2, μ, we have

$$
\begin{aligned}
[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu) &= \mathbb{E}_{\sigma\sim\mu}\{[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\sigma)\}\\
&= \mathbb{E}_{\sigma\sim\mu}\{[\![\langle C_1\rangle]\!](\sigma) \oplus_p [\![\langle C_2\rangle]\!](\sigma)\} \qquad \text{(by Lem. 364)}\\
&= \mathbb{E}_{\sigma\sim\mu}\{[\![\langle C_1\rangle]\!](\sigma)\} \oplus_p \mathbb{E}_{\sigma\sim\mu}\{[\![\langle C_2\rangle]\!](\sigma)\} \qquad \text{(by Lem. 16)}\\
&= [\![\langle C_1\rangle]\!](\mu) \oplus_p [\![\langle C_2\rangle]\!](\mu).
\end{aligned}
$$

Lemma 366 (Soundness of (SQ-PCH) rule). *For all C_1, C_2, P, Q_1, Q_2, G, if $G \models_{SQ} \{P\}C_1\{Q_1\}$ and $G \models_{SQ} \{P\}C_2\{Q_2\}$, then $G \models_{SQ} \{P\}\langle C_1\rangle \oplus_p \langle C_2\rangle\{Q_1 \oplus_p Q_2\}$.*

Proof. For all C_1, C_2, P, Q_1, Q_2, G such that $G \models_{\text{SQ}} \{P\}C_1\{Q_1\}$ and $G \models_{\text{SQ}} \{P\}C_2\{Q_2\}$, to prove $G \models_{\text{SQ}} \{P\}\langle C\rangle\{Q_1 \oplus_p Q_2\}$, we need to prove for all μ, if $\mu \models P$ and $|[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu)| = 1$, then $[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu) \models Q$ and $(\sigma, \sigma') \models G$ for all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\sigma))$. For all μ such that $\mu \models P$ and $|[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu)| = 1$, by Lem. 365 we know $[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu) = [\![\langle C_1\rangle]\!](\mu) \oplus_p [\![\langle C_2\rangle]\!](\mu)$. There are three cases: $p = 0$, $p = 1$ or $0 < p < 1$. We prove the three cases respectively.

- $p = 0$.
  $[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu) = [\![\langle C_1\rangle]\!](\mu) \oplus_p [\![\langle C_2\rangle]\!](\mu) = [\![\langle C_2\rangle]\!](\mu)$. From $|[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu)| = 1$ we know $|[\![\langle C_2\rangle]\!](\mu)| = 1$. From $G \models_{\text{SQ}} \{P\}\langle C_2\rangle\{Q_2\}$, $\mu \models P$ and $|[\![\langle C_2\rangle]\!](\mu)| = 1$ we know $[\![\langle C_2\rangle]\!](\mu) \models Q_2$, thus $[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu) \models Q_2$. From $p = 0$ we know $[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu) \models Q_1 \oplus_p Q_2$. For all σ and σ' such that $\sigma \in \mathit{supp}(\mu)$ and $\sigma' \in \mathit{supp}([\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\sigma))$, by Lem. 364 we know $[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\sigma) = [\![\langle C_1\rangle]\!](\sigma) \oplus_p [\![\langle C_2\rangle]\!](\sigma) = [\![\langle C_2\rangle]\!](\sigma)$. From $\sigma' \in \mathit{supp}([\![\langle C_1\rangle \oplus_p$

$\langle C_2\rangle]\!](\sigma))$ we know $\sigma' \in supp([\![\langle C_2\rangle]\!](\sigma))$. From $G \models_{\text{SQ}} \{P\}\langle C_2\rangle\{Q_2\}$, $\mu \models P$, $|[\![\langle C_2\rangle]\!](\mu)| = 1$, $\sigma \in supp(\mu)$ and $\sigma' \in supp([\![\langle C_2\rangle]\!](\sigma))$ we know $(\sigma, \sigma') \models G$.

- $p = 1$.
  $[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu) = [\![\langle C_1\rangle]\!](\mu) \oplus_p [\![\langle C_2\rangle]\!](\mu) = [\![\langle C_1\rangle]\!](\mu)$. From $|[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu)| = 1$ we know $|[\![\langle C_1\rangle]\!](\mu)| = 1$. From $G \models_{\text{SQ}} \{P\}\langle C_1\rangle\{Q_1\}$, $\mu \models P$ and $|[\![\langle C_1\rangle]\!](\mu)| = 1$ we know $[\![\langle C_1\rangle]\!](\mu) \models Q_1$, thus $[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu) \models Q_1$. From $p = 1$ we know $[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu) \models Q_1 \oplus_p Q_2$. For all σ and σ' such that $\sigma \in supp(\mu)$ and $\sigma' \in supp([\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\sigma))$, by Lem. 364 we know $[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\sigma) = [\![\langle C_1\rangle]\!](\sigma) \oplus_p [\![\langle C_2\rangle]\!](\sigma) = [\![\langle C_1\rangle]\!](\sigma)$. From $\sigma' \in supp([\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\sigma))$ we know $\sigma' \in supp([\![\langle C_1\rangle]\!](\sigma))$. From $G \models_{\text{SQ}} \{P\}\langle C_1\rangle\{Q_1\}$, $\mu \models P$, $|[\![\langle C_1\rangle]\!](\mu)| = 1$, $\sigma \in supp(\mu)$ and $\sigma' \in supp([\![\langle C_1\rangle]\!](\sigma))$ we know $(\sigma, \sigma') \models G$.
- $0 < p < 1$.
  From $|[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu)| = 1$ we know $|[\![\langle C_1\rangle]\!](\mu) \oplus_p [\![\langle C_2\rangle]\!](\mu)| = 1$. From $0 < p < 1$ by Lem. 314 we know $|[\![\langle C_1\rangle]\!](\mu)| = 1$ and $|[\![\langle C_2\rangle]\!](\mu)| = 1$. From $G \models_{\text{SQ}} \{P\}\langle C_1\rangle\{Q_1\}$, $\mu \models P$ and $|[\![\langle C_1\rangle]\!](\mu)| = 1$ we know $[\![\langle C_1\rangle]\!](\mu) \models Q_1$. From $G \models_{\text{SQ}} \{P\}\langle C_2\rangle\{Q_2\}$, $\mu \models P$ and $|[\![\langle C_2\rangle]\!](\mu)| = 1$ we know $[\![\langle C_2\rangle]\!](\mu) \models Q_2$. From $[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu) = [\![\langle C_1\rangle]\!](\mu) \oplus_p [\![\langle C_2\rangle]\!](\mu)$, $0 < p < 1$, $[\![\langle C_1\rangle]\!](\mu) \models Q_1$ and $[\![\langle C_2\rangle]\!](\mu) \models Q_2$ we know $[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\mu) \models Q_1 \oplus_p Q_2$. For all σ and σ' such that $\sigma \in supp(\mu)$ and $\sigma' \in supp([\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\sigma))$, by Lem. 364 we know $[\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\sigma) = [\![\langle C_1\rangle]\!](\sigma) \oplus_p [\![\langle C_2\rangle]\!](\sigma)$. From $0 < p < 1$ by Lem. 275 we know $supp([\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\sigma)) = supp([\![\langle C_1\rangle]\!](\sigma)) \cup supp([\![\langle C_2\rangle]\!](\sigma))$. From $\sigma' \in supp([\![\langle C_1\rangle \oplus_p \langle C_2\rangle]\!](\sigma))$ we know $\sigma' \in supp([\![\langle C_1\rangle]\!](\sigma))$ or $\sigma' \in supp([\![\langle C_2\rangle]\!](\sigma))$. If $\sigma' \in supp([\![\langle C_1\rangle]\!](\sigma))$, from $G \models_{\text{SQ}} \{P\}\langle C_1\rangle\{Q_1\}$, $\mu \models P$, $|[\![\langle C_1\rangle]\!](\mu)| = 1$, $\sigma \in supp(\mu)$ and $\sigma' \in supp([\![\langle C_1\rangle]\!](\sigma))$ we know $(\sigma, \sigma') \models G$. If $\sigma' \in supp([\![\langle C_2\rangle]\!](\sigma))$, from $G \models_{\text{SQ}} \{P\}\langle C_2\rangle\{Q_2\}$, $\mu \models P$, $|[\![\langle C_2\rangle]\!](\mu)| = 1$, $\sigma \in supp(\mu)$ and $\sigma' \in supp([\![\langle C_2\rangle]\!](\sigma))$ we know $(\sigma, \sigma') \models G$.

Lemma 367. *For all C, P, Q, G, if $G \vdash_{\text{SQ}} \{P\}C\{Q\}$, then $G \models_{\text{SQ}} \{P\}C\{Q\}$.*

Proof. For all C, P, Q, G such that $G \vdash_{\text{SQ}} \{P\}C\{Q\}$, we prove $G \models_{\text{SQ}} \{P\}C\{Q\}$ by induction on the derivation of $G \vdash_{\text{SQ}} \{P\}C\{Q\}$.

- case (SQ-DISJ): $P = P_1 \vee P_2$, $Q = Q_1 \vee Q_2$, $G \vdash_{\text{SQ}} \{P_1\}C\{Q_1\}$ and $G \vdash_{\text{SQ}} \{P_2\}C\{Q_2\}$.
  From $G \vdash_{\text{SQ}} \{P_1\}C\{Q_1\}$ by induction hypothesis we know $G \models_{\text{SQ}} \{P_1\}C\{Q_1\}$.
  From $G \vdash_{\text{SQ}} \{P_2\}C\{Q_2\}$ by induction hypothesis we know $G \models_{\text{SQ}} \{P_2\}C\{Q_2\}$.
  By Lem. 305 we know $G \models_{\text{SQ}} \{P_1 \vee P_2\}C\{Q_1 \vee Q_2\}$, i.e., $G \models_{\text{SQ}} \{P\}C\{Q\}$.
- case (SQ-CONJ): $P = P_1 \vee P_2$, $Q = Q_1 \vee Q_2$, $G \vdash_{\text{SQ}} \{P_1\}C\{Q_1\}$ and $G \vdash_{\text{SQ}} \{P_2\}C\{Q_2\}$.
  From $G \vdash_{\text{SQ}} \{P_1\}C\{Q_1\}$ by induction hypothesis we know $G \models_{\text{SQ}} \{P_1\}C\{Q_1\}$.
  From $G \vdash_{\text{SQ}} \{P_2\}C\{Q_2\}$ by induction hypothesis we know $G \models_{\text{SQ}} \{P_2\}C\{Q_2\}$.
  By Lem. 306 we know $G \models_{\text{SQ}} \{P_1 \wedge P_2\}C\{Q_1 \wedge Q_2\}$, i.e., $G \models_{\text{SQ}} \{P\}C\{Q\}$.
- case (SQ-EXIST): $P = \exists X.P_1$, $Q = \exists X.Q_1$, $G \vdash_{\text{SQ}} \{P_1\}C\{Q_1\}$ and $X \notin fv(G) \cup wv(C)$.
  From $G \vdash_{\text{SQ}} \{P_1\}C\{Q_1\}$ by induction hypothesis we know $G \models_{\text{SQ}} \{P_1\}C\{Q_1\}$.
  From $X \notin fv(G) \cup wv(C)$ by Lem. 310 we know $G \models_{\text{SQ}} \{\exists X.P_1\}C\{\exists X.Q\}$, i.e., $G \models_{\text{SQ}} \{P\}C\{Q\}$.

- case (SQ-FORALL): $P = \forall X.P_1$, $Q = \forall X.Q_1$, $G \vdash_{\text{SQ}} \{P_1\}C\{Q_1\}$ and $X \notin fv(G) \cup wv(C)$.
  From $G \vdash_{\text{SQ}} \{P_1\}C\{Q_1\}$ by induction hypothesis we know $G \models_{\text{SQ}} \{P_1\}C\{Q_1\}$.
  From $X \notin fv(G) \cup wv(C)$ by Lem. 311 we know $G \models_{\text{SQ}} \{\forall X.P_1\}C\{\forall X.Q\}$, i.e., $G \models_{\text{SQ}} \{P\}C\{Q\}$.
- case (SQ-CSQ): $P \Rightarrow P'$, $G' \vdash_{\text{SQ}} \{P'\}C\{Q'\}$, $Q' \Rightarrow Q$ and $G' \Rightarrow G$.
  From $G' \vdash_{\text{SQ}} \{P'\}C\{Q'\}$ by induction hypothesis we know $G' \models_{\text{SQ}} \{P'\}C\{Q'\}$.
  From $P \Rightarrow P'$, $Q' \Rightarrow Q$ and $G' \Rightarrow G$ by Lem. 312 we know $G \models_{\text{SQ}} \{P\}C\{Q\}$.
- case (SQ-OPLUS): $P = P_1 \oplus_p P_2$, $Q = Q_1 \oplus_p Q_2$, $G \vdash_{\text{SQ}} \{P_1\}C\{Q_1\}$ and $G \vdash_{\text{SQ}} \{P_2\}C\{Q_2\}$.
  From $G \vdash_{\text{SQ}} \{P_1\}C\{Q_1\}$ by induction hypothesis we know $G \models_{\text{SQ}} \{P_1\}C\{Q_1\}$.
  From $G \vdash_{\text{SQ}} \{P_2\}C\{Q_2\}$ by induction hypothesis we know $G \models_{\text{SQ}} \{P_2\}C\{Q_2\}$.
  By Lem. 315 we know $G \models_{\text{SQ}} \{P_1 \oplus_p P_2\}C\{Q_1 \oplus_p Q_2\}$, i.e., $G \models_{\text{SQ}} \{P\}C\{Q\}$.
- case (SQ-BIGOPLUS): $P = \bigoplus P_1$, $Q = \bigoplus Q_1$ and $G \vdash_{\text{SQ}} \{P_1\}C\{Q_1\}$.
  From $G \vdash_{\text{SQ}} \{P_1\}C\{Q_1\}$ by induction hypothesis we know $G \models_{\text{SQ}} \{P_1\}C\{Q_1\}$.
  By Lem. 320 we know $G \models_{\text{SQ}} \{\bigoplus P_1\}C\{\bigoplus Q\}$, i.e., $G \models_{\text{SQ}} \{P\}C\{Q\}$.
- case (SQ-SKIP): $C = \textbf{skip}$, $P = Q$, $G = \textbf{Id}$.
  By Lem. 325 we know $\textbf{Id} \models_{\text{SQ}} \{Q\}\textbf{skip}\{Q\}$, i.e., $G \models_{\text{SQ}} \{P\}C\{Q\}$.
- case (SQ-SEQ): $C = C_1; C_2$, $G = G_1 \circ G_2$, $G_1 \vdash_{\text{SQ}} \{P\}C_1\{M\}$ and $G_2 \vdash_{\text{SQ}} \{M\}C_2\{Q\}$.
  From $G_1 \vdash_{\text{SQ}} \{P\}C_1\{M\}$ by induction hypothesis we know $G_1 \models_{\text{SQ}} \{P\}C_1\{M\}$.
  From $G_2 \vdash_{\text{SQ}} \{M\}C_2\{Q\}$ by induction hypothesis we know $G_2 \models_{\text{SQ}} \{M\}C_2\{Q\}$.
  By Lem. 330 we know $G_1 \circ G_2 \models_{\text{SQ}} \{P\}C_1; C_2\{Q\}$, i.e., $G \models_{\text{SQ}} \{P\}C\{Q\}$.
- case (SQ-ASGN): $C = x := e$, $P \Rightarrow Q[e/x]$ and $(\sigma, \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}) \models G$ for all σ and μ such that $\sigma \in \mathit{supp}(\mu)$ and $\mu \models P$.
  From $P \Rightarrow Q[e/x]$ and $(\sigma, \sigma\{x \rightsquigarrow [\![e]\!]_\sigma\}) \models G$ for all σ and μ such that $\sigma \in \mathit{supp}(\mu)$ and $\mu \models P$ by Lem. 353 we know $G \models_{\text{SQ}} \{P\}x := e\{Q\}$, i.e., $G \models_{\text{SQ}} \{P\}C\{Q\}$.
- case (SQ-COND): $C = \textbf{if}\ (b)\ \textbf{then}\ C_1\ \textbf{else}\ C_2$, $P = (P_1 \wedge \lceil b \rceil) \oplus_p (P_2 \wedge \lceil \neg b \rceil)$, $Q = Q_1 \oplus_p Q_2$, $G \vdash_{\text{SQ}} \{P_1 \wedge \lceil b \rceil\}C_1\{Q_1\}$ and $G \vdash_{\text{SQ}} \{P_2 \wedge \lceil \neg b \rceil\}C_2\{Q_2\}$.
  From $G \vdash_{\text{SQ}} \{P_1 \wedge \lceil b \rceil\}C_1\{Q_1\}$ by induction hypothesis we know $G \models_{\text{SQ}} \{P_1 \wedge \lceil b \rceil\}C_1\{Q_1\}$. From $G \vdash_{\text{SQ}} \{P_2 \wedge \lceil \neg b \rceil\}C_2\{Q_2\}$ by induction hypothesis we know $G \models_{\text{SQ}} \{P_2 \wedge \lceil \neg b \rceil\}C_2\{Q_2\}$. By Lem. 360 we know $G \models_{\text{SQ}} \{(P_1 \wedge \lceil b \rceil) \oplus_p (P_2 \wedge \lceil \neg b \rceil)\}\textbf{if}\ (b)\ \textbf{then}\ C_1\ \textbf{else}\ C_2\{Q_1 \oplus_p Q_2\}$, i.e., $G \models_{\text{SQ}} \{P\}C\{Q\}$.
- case (SQ-ATOM): $C = \langle C_1 \rangle$ and $G \vdash_{\text{SQ}} \{P\}C_1\{Q\}$.
  From $G \vdash_{\text{SQ}} \{P\}C_1\{Q\}$ by induction hypothesis we know $G \models_{\text{SQ}} \{P\}C_1\{Q\}$.
  By Lem. 362 we know $G \models_{\text{SQ}} \{P\}\langle C_1 \rangle\{Q\}$, i.e., $G \models_{\text{SQ}} \{P\}C\{Q\}$.
- case (SQ-PCH): $C = \langle C_1 \rangle \oplus_p \langle C_2 \rangle$, $Q = Q_1 \oplus_p Q_2$, $G \vdash_{\text{SQ}} \{P\}\langle C_1 \rangle\{Q_1\}$ and $G \vdash_{\text{SQ}} \{P\}\langle C_2 \rangle\{Q_2\}$.
  From $G \vdash_{\text{SQ}} \{P\}\langle C_1 \rangle\{Q_1\}$ by induction hypothesis we know $G \models_{\text{SQ}} \{P\}\langle C_1 \rangle\{Q_1\}$.
  From $G \vdash_{\text{SQ}} \{P\}\langle C_2 \rangle\{Q_2\}$ by induction hypothesis we know $G \models_{\text{SQ}} \{P\}\langle C_2 \rangle\{Q_2\}$.
  By Lem. 366 we know $G \models_{\text{SQ}} \{P\}\langle C_1 \rangle \oplus_p \langle C_2 \rangle\{Q_1 \oplus_p Q_2\}$, i.e., $G \models_{\text{SQ}} \{P\}C\{Q\}$.