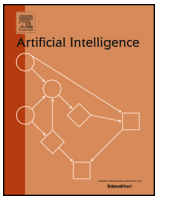
Artificial Intelligence 326 (2024) 104045

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



Sound and relatively complete belief Hoare logic for statistical hypothesis testing programs

Yusuke Kawamoto [a,b,*], Tetsuya Sato [c], Kohei Suenaga [d]

[a] AIST, Tokyo, Japan
[b] PRESTO, JST, Japan
[c] Tokyo Institute of Technology, Tokyo, Japan
[d] Kyoto University, Kyoto, Japan

ARTICLE INFO

Keywords:
Knowledge representation
Epistemic logic
Program logic
Kripke model
Statistical hypothesis testing

ABSTRACT

We propose a new approach to formally describing the requirement for statistical inference and checking whether a program uses the statistical method appropriately. Specifically, we define belief Hoare logic (BHL) for formalizing and reasoning about the statistical beliefs acquired via hypothesis testing. This program logic is sound and relatively complete with respect to a Kripke model for hypothesis tests. We demonstrate by examples that BHL is useful for reasoning about practical issues in hypothesis testing. In our framework, we clarify the importance of prior beliefs in acquiring statistical beliefs through hypothesis testing, and discuss the whole picture of the justification of statistical inference inside and outside the program logic.

1. Introduction

Statistical inferences have been increasingly used to derive and justify scientific knowledge in a variety of academic disciplines, from natural sciences to social sciences. This has significantly raised the importance of statistics, but also brought concerns about the inappropriate procedure and the incorrect interpretation of statistics in scientific research. Notably, previous studies have pointed out that many research articles in biomedical science contain severe errors in applying statistical methods and interpreting their outcomes [1]. Furthermore, large proportions of these errors have been reported for basic statistical methods, possibly performed by researchers who can use only elementary techniques. In particular, the concept of *statistical significance*, evaluated using *p-values*, has been commonly misused and misinterpreted [2].

Key factors underlying these human errors are that (i) the requirements for statistical inference are typically implicit or unrecognized, and that (ii) the logical aspects of statistical inference are described informally in natural language and handled manually by analysts who may not fully understand the statistical methods. As a result, analysts may overlook some of the assumptions necessary for statistical methods, hence choosing inappropriate methods. Nevertheless, to our knowledge, no prior work on formal methods has specified the preconditions for statistical inference programs or verified the choice of statistical techniques in programs.

In this paper, we propose a method for formalizing and reasoning about statistical inference using symbolic logic. Specifically, we introduce sound and relatively complete *belief Hoare logic* (*BHL*) to formalize the statistical beliefs acquired via hypothesis tests,

* Corresponding author at: AIST, Tokyo, Japan.
E-mail address: yusuke.kawamoto@aist.go.jp (Y. Kawamoto).

https://doi.org/10.1016/j.artint.2023.104045
Received 15 August 2022; Received in revised form 25 October 2023; Accepted 31 October 2023

Available online 10 November 2023

and to prevent errors in the choice of hypothesis tests by describing their preconditions explicitly. We demonstrate by examples that this logic can be used to reason about practical issues concerning statistical inference.

1.1. Contributions

Our main contributions are as follows:

- We propose a new approach to formalizing and reasoning about statistical inference in a program. In particular, this approach formalizes and checks the requirement for statistical methods to be used appropriately in a program.
- We define an epistemic language to express statistical beliefs obtained by hypothesis tests on datasets. Specifically, we formalize a statistical belief in a hypothesis φ as the knowledge that either (i) φ holds, (ii) the sampled dataset is unluckily far from the population, or (iii) the population does not satisfy the requirements for the hypothesis test. Then we introduce a Kripke model for hypothesis tests to define the interpretation of this language.
- Using this epistemic language, we construct belief Hoare logic (BHL) for reasoning about statistical hypothesis testing programs. Then we prove that BHL is sound and relatively complete w.r.t. the Kripke model for hypothesis tests.
- We clarify the importance of prior beliefs in acquiring statistical beliefs, and prove essential properties of statistical beliefs by using our framework.
- We show that BHL is useful for reasoning about practical issues concerning statistical inference, such as p-value hacking and multiple comparison problems.
- We provide the whole picture of the justification of statistical beliefs acquired via hypothesis tests inside and outside BHL. In particular, we discuss the empirical conditions for hypothesis tests and the epistemic aspects of statistical inference.

To the best of our knowledge, this appears to be the first attempt to introduce a program logic that can specify the requirements for hypothesis tests to be applied appropriately. We consider this as the first step to building a framework for formalizing and verifying the validity of empirical science and data-driven artificial intelligence.

1.2. Relation with the preliminary version

A preliminary version of this work, considering only sound but *not* complete program logic, appeared in [3]. The main novelties of this paper to that version are:

- We introduce a sound and *relatively complete* belief Hoare logic (BHL) that has a simpler set of axioms and inference rules than our preliminary version [3].
- We extend the notion of a possible world with a hypothesis test history, and redefine the assertion language. This enables us to provide a more rigorous model for statistical beliefs and to prove the relative completeness of BHL.
- We add propositions and discussion for hypothesis formulas and show the importance of prior beliefs in hypothesis testing by using our framework in Section 7.
- We present all proofs for our technical results in Appendix B.

1.3. Related work

Hoare logic [4,5] is a form of program logic for an imperative programming language. This program logic is then extended and adapted so that it can handle various types of programs and assertions [6], including heap-manipulating programs [7], hybrid systems [8], and probabilistic programs [9]. Atkinson and Carbin propose an extension of Hoare logic with epistemic assertions [10]. In their work, an epistemic assertion is used to reason about the belief of a program about a partially observable environment, whereas their logic does not deal with a belief arising from statistical tests conducted in a program. To the best of our knowledge, ours appears to be the first program logic that formalizes the concept of statistical beliefs in hypothesis testing.

Epistemic logic [11,12] is a branch of modal logic for reasoning about knowledge and belief [13]. It has been used to specify and verify various knowledge properties in systems, e.g., authentication [14] and anonymity [15,16]. Many previous works on epistemic logic incorporate probabilistic notions of beliefs [17] and certain notions of degrees of beliefs and confidence [18]. Notably, Bacchus et al. [19] define the degree of belief in a possible world semantics where each world is associated with a weight and the degree of belief in a formula φ is defined as the normalized sum of the weights of all accessible possible worlds satisfying φ. However, this line of studies has not modeled the degree of belief in the sense of statistical significance in a hypothesis test. In contrast, our framework models the degree of belief in terms of a p-value without assigning a weight to a possible world.

The first attempt to express statistical properties of hypothesis tests using modal logic is the work on statistical epistemic logic (StatEL) [20–22]. They introduce a belief modality weaker than S5, and a Kripke model with an accessibility relation defined using a statistical distance between possible worlds. Unlike our work, however, StatEL cannot describe the procedures of statistical methods or reason about their correctness.

Dynamic epistemic logic (DEL) [23] is a branch of modal logic for reasoning about the changes of knowledge and belief when events take place. For a precondition ψ, a postcondition φ, and a terminating program C, a DEL formula $\psi \to [C]\varphi$ with a single

agent expresses a Hoare triple $\{\psi\}\ C\ \{\varphi\}$. Therefore, DEL may be extended to deal with hypothesis testing programs by incorporating the modal operators and predicate symbols for statistical notions introduced in our paper.

Fuzzy logic [24] is a branch of many-valued logic where the truth values range over $[0,1]$. It has been used to model and reason about the degrees of uncertainty in beliefs and confidence [25,26]. To the best of our knowledge, however, no prior work on fuzzy logic can reason about the correct application of statistical hypothesis testing.

Default logic [27] is a branch of logic for *reasoning-by-default*, in which the absence of evidence for an exception leads to a conclusion by default. By extending default logic, a few studies [28,29] have formalized the reasoning in hypothesis tests. In particular, they manage to syntactically deal with the *non-monotonicity* [30] of statistical reasoning to formalize that a conclusion of a hypothesis test may be retracted on the basis of further data. However, since these extensions of default logic do not allow for describing programs, they do not derive the correctness of hypothesis testing programs or methods. Furthermore, these previous studies do not provide a semantics for their default logic based on statistical models and do not prove the soundness of the logic in terms of statistics. In contrast, we prove the soundness and relative completeness of our logic w.r.t. a possible world semantics extended with statistics.

We remark that reasoning-by-default is not necessary to formalize the correctness of hypothesis testing methods in our program logic. This is because, given a (mathematical) statistical model, the requirements for hypothesis testing methods can be formally expressed as assumptions equipped with belief modality in our assertion language, without requiring the notion of *justifications* in default logic. Furthermore, we handle the non-monotonicity of statistical reasoning using our programming language and its operational semantics. Instead of dealing directly with non-monotonic statements about p-values, we introduce a *test history* that grows monotonically with the executions of hypothesis test commands in a Kripke model.

From a broader perspective, many studies formalize and reason about programs based on knowledge [31] and beliefs [32]. For example, Sardina and Lespérance [33] extend the situation calculus-based agent programming language GOLOG [34] with BDI (belief-desire-intention) [35] agents. Belle and Levesque [36] propose a belief-based programming language called ALLEGRO to deal with the probabilistic degrees of beliefs in programs with noisy acting and sensing. However, no prior work appears to have studied belief-based programs involving statistical hypothesis testing.

1.4. Plan of the paper

In Section 2, we review fundamental concepts from statistical hypothesis testing. In Section 3, we present an illustrating example to explain the basic ideas of our framework. In Section 4, we introduce a Kripke model for describing statistical properties and define hypothesis testing. In Section 5, we introduce the syntax and the semantics of an imperative programming language Prog. In Section 6, we define an assertion language, called *epistemic language for hypothesis testing* (ELHT), that can express statistical beliefs. In Section 7, we clarify the importance of prior beliefs in acquiring statistical beliefs, and show the essential properties of statistical beliefs in our framework. In Section 8, we introduce *belief Hoare logic* (BHL) for formalizing and reasoning about statistical inference using hypothesis tests. Then we show the soundness and relative completeness of BHL. In Section 9, we apply our framework to the reasoning about *p-value hacking* and *multiple comparison problems* using BHL. In Section 10, we provide the whole picture of the justification of statistical beliefs inside and outside BHL. In Section 11, we present our final remarks.

In Appendix A, we present examples of the instantiations of derived rules with concrete hypothesis testing methods. In Appendix B, we show the proofs for the propositions about assertions, basic results on structural operational semantics, remarks on parallel compositions, and the proofs for BHL's soundness and relative completeness.

2. Preliminaries

In this section, we introduce notations used in this paper and recall background on statistical hypothesis testing [37,38].

Let $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}_{\geq 0}$ be the sets of non-negative integers, real numbers, and non-negative real numbers, respectively. Let $[0,1]$ be the set of non-negative real numbers less than or equal to 1. We denote the set of all finite vectors of elements in S by S^*, the *set of all multisets of elements in* S *by* $\mathcal{P}(S)$, and the *set of all probability distributions* over a set S by $\mathbb{D}S$. Given two distributions $D_1 \in \mathbb{D}S_1$ and $D_2 \in \mathbb{D}S_2$, a *coupling* of D_1 and D_2 is a joint distribution $D \in \mathbb{D}(S_1 \times S_2)$ whose marginal distributions $\sum_{s_2 \in S_2} D[s_1, s_2]$ and $\sum_{s_1 \in S_1} D[s_1, s_2]$ are identical to D_1 and D_2, respectively.

Statistical hypothesis testing is a method of statistical inference about an unknown *population* x (the collection of items of interest) on the basis of a *dataset* y sampled from x. In a hypothesis test, an *alternative hypothesis* φ_1 is a proposition that we wish to prove about the population x, and a *null hypothesis* φ_0 is a proposition that contradicts φ_1. The goal of the hypothesis test is to determine whether we *accept* the alternative hypothesis φ_1 by *rejecting* the null hypothesis φ_0.

In a hypothesis test, we calculate a *test statistic* $t(y)$ from a dataset y, and see whether the $t(y)$ value contradicts the assumption that the null hypothesis φ_0 is true. Specifically, we calculate the *p-value*, showing the degree of likeliness of obtaining $t(y)$ when the null hypothesis φ_0 is true. If the p-value is smaller than a threshold (e.g., 0.05), we regard the dataset y is unlikely to be sampled from the population satisfying the null hypothesis φ_0, hence we reject φ_0 and accept the alternative hypothesis φ_1.

A hypothesis test is based on a *statistical model* $P(\xi, \theta)$ with unknown parameters ξ, known parameters θ, and (assumed) probability distributions of the parameters ξ.

Example 1 *(Z-test for two population means).* As an illustrating example, we present the *two-tailed Z-test* for means of two populations. We introduce its statistical model as two normal distributions $N(\mu_1, \sigma^2)$ and $N(\mu_2, \sigma^2)$ with a known variance σ^2 and unknown true means μ_1, μ_2. Let y_1 and y_2 be two given datasets where each data value was sampled from $N(\mu_1, \sigma^2)$ and $N(\mu_2, \sigma^2)$, respectively.

Table 1
Hypotheses in the Z-tests.

Tails	prior knowledge	alternative hypothesis φ_1	null hypothesis φ_0
Two	nothing	$\mu_1 \neq \mu_2$	$\mu_1 = \mu_2$
Upper	$\mu_1 \geq \mu_2$	$\mu_1 > \mu_2$	$\mu_1 = \mu_2$
Lower	$\mu_1 \leq \mu_2$	$\mu_1 < \mu_2$	$\mu_1 = \mu_2$

In the Z-test, we wish to prove the alternative hypothesis $\varphi_1 \stackrel{\text{def}}{=} (\mu_1 \neq \mu_2)$ by rejecting the null hypothesis $\varphi_0 \stackrel{\text{def}}{=} (\mu_1 = \mu_2)$. First, we calculate the Z-test statistic:

$$t(y_1, y_2) = \frac{\mathsf{mean}(y_1) - \mathsf{mean}(y_2)}{\sigma \sqrt{1/\mathsf{size}(y_1) + 1/\mathsf{size}(y_2)}}$$

where for $b = 1, 2$, $\mathsf{size}(y_b)$ is the sample size of the dataset y_b and $\mathsf{mean}(y_b)$ is the mean of all data in y_b. Then we calculate the *p-value*

$$\Pr_{(d_1,d_2)\sim N(\mu_1,\sigma^2)\times N(\mu_1,\sigma^2)} [\, |t(d_1, d_2)| > |t(y_1, y_2)| \,]$$

under the null hypothesis φ_0. When the p-value is small enough, the datasets y_1 and y_2 are unlikely to be sampled from the same distribution, i.e., the null hypothesis $\mu_1 = \mu_2$ is unlikely to hold. Hence, in the Z-test, if the p-value is smaller than a certain threshold (e.g., 0.05), we reject the null hypothesis φ_0 and accept the alternative hypothesis φ_1.

When we have prior knowledge of $\mu_1 \geq \mu_2$ (resp. $\mu_1 \leq \mu_2$), then we apply the *upper-tailed* (resp. *lower-tailed*) Z-test with the alternative hypothesis $\mu_1 > \mu_2$ (resp. $\mu_1 < \mu_2$) and the null hypothesis $\mu_1 = \mu_2$, and obtain the p-value $\Pr[t(d_1, d_2) > t(y_1, y_2)]$ (resp. $\Pr[t(d_1, d_2) < t(y_1, y_2)]$). In Table 1, we summarize the hypotheses of the Z-tests with different tails.

For more details, the readers are referred to standard textbooks, e.g., [37,38].

3. Illustrating example

Throughout the paper, we use the following simple illustrating example to explain the basic ideas of our framework.

Example 2 *(Comparison tests on drugs).* Let us consider three drugs 1, 2, 3 that may decrease blood pressure. To compare the efficacy of these drugs, we perform experiments and obtain a set y_i of the reduced values of blood pressure after taking drug i. Then we apply hypothesis tests on the dataset $y = (y_1, y_2, y_3)$. We assume that the data values in y_i have been sampled from the population that follows a normal distribution $N(\mu_i, \sigma^2)$ with a mean μ_i and a variance σ^2. For simplicity, we consider the situation where we know the variance σ^2 but do not know the means μ_i.

Suppose that drug 1 is composed of drugs 2 and 3, and we investigate whether drug 1 has better efficacy than *both* drugs 2 and 3. Then we take the following procedure:

- We first compare drugs 1 and 2 concerning the average decreases in blood pressure. We apply a two-tailed Z-test A_{12} (Example 1) to see whether the means of the populations are different, i.e., $\mu_1 \neq \mu_2$. In this test, the alternative hypothesis φ_{12} is the inequality $\mu_1 \neq \mu_2$, and the null hypothesis $\neg\varphi_{12}$ is $\mu_1 = \mu_2$.
- Let α_{ij} be the p-value when only comparing drugs i and j.
- If $\alpha_{12} \geq 0.05$, the Z-test A_{12} does not reject the null hypothesis $\neg\varphi_{12}$ and concludes that the efficacy of drugs 1 and 2 may be the same. Then we are not interested in drug 1 any more, and skip the comparison with drug 3.
- If $\alpha_{12} < 0.05$, the Z-test A_{12} rejects the null hypothesis $\neg\varphi_{12}$ and concludes that the alternative hypothesis φ_{12} is true. Then we apply another Z-test A_{13} to check whether the alternative hypothesis $\varphi_{13} \stackrel{\text{def}}{=} (\mu_1 \neq \mu_3)$ is true.
- Finally, we calculate the p-value of the combined test A consisting of A_{12} and A_{13}, with the *conjunctive* alternative hypothesis $\varphi_{12} \wedge \varphi_{13}$.

Overview of the Framework. In our framework, we describe a procedure of statistical tests as a program using a programming language (Section 5); in Example 2, we denote the Z-test program comparing drugs i with j by C_{ij}, and the whole procedure by:

$$C_{\text{drug}} \stackrel{\text{def}}{=} C_{12}; \texttt{if}\ \alpha_{12} < 0.05\ \texttt{then}\ C_{13}\ \texttt{else}\ \texttt{skip} \tag{1}$$

Then we use an assertion logic (Section 6) to describe the requirement for the hypothesis tests as a *precondition* formula. In Example 2, the precondition is given by:

$$\psi_{\text{pre}} \stackrel{\text{def}}{=} \bigwedge_{i=1,2,3} y_i \underset{n_i}{\leftsquigarrow} N(\mu_i, \sigma^2) \wedge \mathbf{P}(\varphi_{12} \wedge \varphi_{13}) \wedge \kappa_{\emptyset}.$$

In this formula, $y_i \underset{n_i}{\leftsquigarrow} N(\mu_i, \sigma^2)$ represents that a set y_i of n_i data is sampled from the population that follows the normal distribution $N(\mu_i, \sigma^2)$. The modal formula $\mathbf{P}(\varphi_{12} \wedge \varphi_{13})$ represents that before conducting the hypothesis tests, we have the *prior belief* that the

alternative hypothesis $\varphi_{12} \wedge \varphi_{13}$ *may be* true (see Section 7 for discussion). The formula $\kappa_{\emptyset}$ represents that no hypothesis test has been conducted previously.

The statistical belief we want to acquire is specified as a *postcondition* formula. In Example 2, the postcondition is:

$$\varphi_{\mathsf{post}} \stackrel{\text{def}}{=} \mathbf{K}^{\leq 0.05}_{y,A_{12}} \varphi_{12} \to \mathbf{K}^{\leq \min(\alpha_{12},\alpha_{13})}_{y,A}(\varphi_{12} \wedge \varphi_{13}). \tag{2}$$

Intuitively, by testing on the dataset y, when we believe φ_{12} with a p-value $\alpha \leq 0.05$, we believe the combined hypothesis $\varphi_{12} \wedge \varphi_{13}$ with a p-value at most $\min(\alpha_{12}, \alpha_{13})$.

Finally, we combine all the above and describe the whole statistical inference as a *judgment*. In Example 2, we write:

$$\Gamma \vdash \{\psi_{\mathsf{pre}}\}\ C_{\mathrm{drug}}\ \{\varphi_{\mathsf{post}}\}. \tag{3}$$

By proving this judgment using rules in BHL (Section 8), we conclude that the statistical inference is appropriate.

We remark that the p-value can be larger for a different purpose of testing. Suppose that in Example 2, drug 1 was a new drug and we wanted to find out that it had better efficacy than *at least one* of drugs 2 and 3. Then the procedure is:

$$C_{\mathrm{multi}} \stackrel{\text{def}}{=} C_{12} \,\|\, C_{13}, \tag{4}$$

and the alternative hypothesis is $\varphi_{12} \vee \varphi_{13}$ with a p-value greater than α_{12} and α_{13} (at most $\alpha_{12} + \alpha_{13}$). This is the *multiple comparisons problem* [39], arising when the combined alternative hypothesis is *disjunctive*. We explain more details in Section 8.

4. Model

In this section, we introduce a Kripke model for describing statistical properties and formally define hypothesis tests.

4.1. Variables, data, and actions

We introduce a finite set Var of variables comprised of two disjoint sets of *invisible variables* and of *observable variables*: $\mathsf{Var} = \mathsf{Var}_{\mathsf{inv}} \cup \mathsf{Var}_{\mathsf{obs}}$. We can directly observe the values of the latter, but not those of the former. Throughout the paper, we use y as an observable variable denoting a dataset sampled from the population.

We write $\mathcal{O}$ for the set of all *data values* that consists of the Boolean values, integers, real numbers, distributions of data values, and lists of data values. A *dataset* is a list of lists of data values. In particular, we deal with a list of real vectors as a dataset. Then the vectors range over $\mathcal{X} \stackrel{\text{def}}{=} \mathbb{R}^l$ for an $l \in \mathbb{N}$. A distribution over a population has type $\mathbb{D}\mathcal{X}$, and a dataset has type $\mathsf{list}\,\mathcal{X}$. We remark that distributions and datasets are elements of $\mathcal{O}$; i.e., $\mathbb{D}\mathcal{X} \subseteq \mathcal{O}$ and $\mathsf{list}\,\mathcal{X} \subseteq \mathcal{O}$. $\bot$ denotes the undefined value.

We write $d \sim D^n$ for the sampling of a set d of n data from a population D where all these data are independent and identically distributed (i.i.d.). Let Smpl be a set of i.i.d. samplings of datasets from populations (e.g., $d \sim D^n$), and Cmd be a set of program commands (e.g., $v := e$ and $\mathtt{skip}$). Then we define an *action* as a sampling of a dataset or a program command; i.e., $\mathsf{Act} = \mathsf{Smpl} \cup \mathsf{Cmd}$. In Section 5, we instantiate Cmd with concrete commands used in a programming language.

4.2. States and possible worlds

We introduce the notions of states and possible worlds equipped with test histories. We write $\mathcal{A}$ for a finite set of hypothesis tests we consider.

Definition 1 *(States).* A *state* is a tuple (m, a, H) consisting of (i) the *current assignment* $m : \mathsf{Var} \to \mathcal{O} \cup \{\bot\}$ of data values to variables, (ii) the action $a \in \mathsf{Act}$ that has been executed in the last transition, and (iii) the *test history* $H : (\mathsf{list}\,\mathcal{X}) \to \mathcal{P}(\mathcal{A})$ that maps a dataset d to the *multiset* of all hypothesis tests that have used the dataset d.

We remark that $H(d)$ is a multiset rather than a set, because the same test on the same dataset d can be performed multiple times.

Definition 2 *(Possible worlds).* A *possible world* w is a sequence of states $(w[0], w[1], \ldots, w[k-1])$ where $w[i]$ is the i-th state in w. $w[0]$ and $w[k-1]$ are called the *initial state* and the *current state*, respectively. The length k is denoted by $\mathsf{len}(w)$. We write (m_w, a_w, H_w) for the current state $w[k-1]$ of a possible world w. We assume that the test history is empty at the initial states.

Since a possible world records all updates of data values, it can be used to model the updates of knowledge and beliefs. As with previous works on epistemic logic [13], agents' knowledge and belief are defined from their *observation* of possible worlds.

Definition 3 *(Observation).* The *observation* of a state $w[i] = (m, a, H)$ is defined by $obs(w[i]) = (m^{\mathsf{obs}}, a, H)$ with an assignment $m^{\mathsf{obs}} : \mathsf{Var}_{\mathsf{obs}} \to \mathcal{O} \cup \{\bot\}$ such that $m^{\mathsf{obs}}(v) = m(v)$ for all $v \in \mathsf{Var}_{\mathsf{obs}}$, and that $m^{\mathsf{obs}}(v) = \bot$ for all $v \in \mathsf{Var}_{\mathsf{inv}}$. The *observation* of a world w is given by $obs(w) = (obs(w[0]), \ldots, obs(w[k-1]))$.

4.3. Kripke model

We introduce a *Kripke model with labeled transitions* where two kinds of relations $\xrightarrow{a}$ and $\mathcal{R}$ may relate possible worlds. A *transition relation* $w \xrightarrow{a} w'$ represents a transition from a world w to another w' by performing an action a. An *observability relation* $w\mathcal{R}w'$ represents that two possible worlds w and w' have the same observation, i.e., $obs(w) = obs(w')$. Then $\mathcal{R}$ is an equivalence relation. Furthermore, for any worlds w and w', $w\mathcal{R}w'$ implies $H_w = H_{w'}$. In Section 6, this relation is used to model *knowledge* in the conventional Hintikka style.

Definition 4 *(Kripke model).* A *Kripke model* is a tuple $\mathfrak{M} = (\mathcal{W}, (\xrightarrow{a})_{a\in\mathsf{Act}}, \mathcal{R}, (V_w)_{w\in\mathcal{W}})$ consisting of:

- a non-empty set $\mathcal{W}$ of possible worlds;
- for each $a \in \mathsf{Act}$, a transition relation $\xrightarrow{a} \subseteq \mathcal{W} \times \mathcal{W}$;
- an observability relation $\mathcal{R} = \{(w, w') \in \mathcal{W} \times \mathcal{W} \mid obs(w) = obs(w')\}$;
- for each $w \in \mathcal{W}$, a valuation V_w that maps a k-ary predicate symbol to a set of k-tuples of data values.

We assume that each world in a model has the same sets $\mathsf{Var}_{\mathsf{inv}}$ and $\mathsf{Var}_{\mathsf{obs}}$ of variables.

In Section 5.2, we instantiate the actions in a Kripke model with concrete program commands described in a programming language, and define the transition relation $\xrightarrow{a}$ as the semantic relation $[\![a]\!]$. For example, in a transition $w \xrightarrow{v:=1} w'$, an assignment action $v := 1$ is executed and the resulting state $(m_{w'}, a_{w'}, H_{w'})$ is $(m_w[v \mapsto 1], v := 1, H_w)$. This is formally defined as $(w, w') \in [\![v := 1]\!]$ in Section 5.2.

Throughout this paper, we deal with a class Pred of predicate symbols whose interpretations are identical in any possible world having the same memory; i.e., for any $\eta \in \mathsf{Pred}$ and any $w, w' \in \mathcal{W}$, if $m_w = m_{w'}$ then $V_w(\eta) = V_{w'}(\eta)$.

4.4. Formulation of hypothesis testing

Next, we formalize the notion of hypothesis tests as follows.

Definition 5 *(Hypothesis tests).* We consider a *basic test type* $s \in \{\mathsf{L}, \mathsf{U}, \mathsf{T}\}$ each representing a *lower-tailed*, *upper-tailed*, and *two-tailed* test. A *hypothesis test* is a tuple $A_{\varphi_0} = (\varphi_0, t, D_{t,\varphi_0}, \preccurlyeq^{(s)}, P)$ consisting of:

- φ_0 is an assertion, called a null hypothesis;
- t is a function that maps a dataset $d \in \mathsf{list}\,\mathcal{X}$ to its test statistic $t(d)$, usually with $\mathsf{range}(t) = \mathbb{R}^k$ for a $k \geq 1$;
- $D_{t,\varphi_0} \in \mathbb{D}(\mathsf{range}(t))$ is a probability distribution of the test statistic when the null hypothesis φ_0 is true;
- $\preccurlyeq^{(s)}_t \in \mathsf{range}(t) \times \mathsf{range}(t)$ is a *likeliness relation* where for a test type s and for values r and r' of the test statistic, $r \preccurlyeq^{(s)}_t r'$ represents that r is at most as likely as r'. For brevity, we often omit t to write $\preccurlyeq^{(s)}$;
- $P(\xi, \theta)$ denotes the population following a statistical model P with unknown parameters ξ and known parameters θ.

For brevity, we abbreviate A_φ as A. We denote by P_A the statistical model P of a hypothesis test A, and by $\mathcal{A}$ a finite set of hypothesis tests we consider.

Example 3 *(The likeliness relation for Z-test).* The two-tailed Z-test for means of two populations in Example 1 can be denoted by the following hypothesis test:

$$A_{\varphi_0} = (\varphi_0, t, N(0,1), \preccurlyeq^{(\mathsf{T})}, N(\mu_1, \sigma^2) \times N(\mu_2, \sigma^2)).$$

The likeliness relation $r \preccurlyeq^{(\mathsf{T})} r'$ expresses $|r| \geq |r'|$. When the null hypothesis φ_0 is true, the test statistic $t(y_1, y_2)$ follows the standard normal distribution $N(0,1)$, hence

$$\Pr[t(y_1, y_2) \preccurlyeq^{(\mathsf{T})} 1.96] = \Pr[|t(y_1, y_2)| \geq 1.96] \approx 0.05.$$

For the upper-tailed (lower-tailed) test, with alternative hypothesis $\varphi_{\mathsf{U}} \stackrel{\text{def}}{=} (\mu_1 > \mu_2)$ (resp. $\varphi_{\mathsf{L}} \stackrel{\text{def}}{=} (\mu_1 < \mu_2)$), the likeliness relation $r \preccurlyeq^{(\mathsf{U})} r'$ (resp. $r \preccurlyeq^{(\mathsf{L})} r'$) is defined by $r \geq r'$ (resp. $r \leq r'$).

Next, we define *disjunctive/conjunctive combinations* of hypothesis tests. Intuitively, a disjunctive combination $A_{\varphi_1 \vee \varphi_2}$ (resp. conjunctive combination $A_{\varphi_1 \wedge \varphi_2}$) is a hypothesis test with a null hypothesis $\varphi_1 \vee \varphi_2$ (resp. $\varphi_1 \wedge \varphi_2$) that performs two hypothesis tests A_{φ_1} and A_{φ_2} in parallel.

Definition 6 *(Combination of tests).* For $b = 1, 2$, let $A_{\varphi_b} = (\varphi_b, t_b, D_{t_b,\varphi_b}, \preccurlyeq^{(s_b)}_{t_b}, P_b)$ be two hypothesis tests. The *disjunctive combination* of A_{φ_1} and A_{φ_2} is given by

$$A_{\varphi_1 \vee \varphi_2} = (\varphi_1 \vee \varphi_2, t, D_{t,(\varphi_1,\varphi_2)}, \preccurlyeq_t^{(s_1,s_2)}, P)$$

where $t(y_1, y_2) = (t_1(y_1), t_2(y_2))$, $D_{t,(\varphi_1,\varphi_2)}$ is a coupling of D_{t_1,φ_1} and D_{t_2,φ_2} (i.e., it is a joint distribution such that D_{t_1,φ_1} and D_{t_2,φ_2} are the marginal distributions of $D_{t,(\varphi_1,\varphi_2)}$), $(r_1, r_2) \preccurlyeq_t^{(s_1,s_2)} (r'_1, r'_2)$ iff either $r_1 \preccurlyeq_{t_1}^{(s_1)} r'_1$ or $r_2 \preccurlyeq_{t_2}^{(s_2)} r'_2$, and P is a coupling of P_1 and P_2. Similarly, the *conjunctive combination* of A_{φ_1} and A_{φ_2} is

$$A_{\varphi_1 \wedge \varphi_2} = (\varphi_1 \wedge \varphi_2, t, D_{t,(\varphi_1,\varphi_2)}, \preccurlyeq_t^{(s_1,s_2)}, P)$$

where $(r_1, r_2) \preccurlyeq_t^{(s_1,s_2)} (r'_1, r'_2)$ iff $r_1 \preccurlyeq_{t_1}^{(s_1)} r'_1$ and $r_2 \preccurlyeq_{t_2}^{(s_2)} r'_2$.

Then we define a function $\ell_{y,A}$ to decompose a combined test into individual tests.

Definition 7. For a combination A of n hypothesis tests $A_1, \ldots, A_n$ and a tuple of n datasets $y = (y_1, \ldots, y_n)$, the multiset of all the pairs of datasets and tests is:

$$\ell_{y,A} \stackrel{\text{def}}{=} \{(y_i, A_i) \mid i = 1, \ldots, n\}. \tag{5}$$

For instance, $\ell_{(y_1,y_2),A_{\varphi_1 \wedge \varphi_2}} = \{(y_1, A_{\varphi_1}), (y_2, A_{\varphi_2})\}$.

5. A simple programming language

In this section, we introduce an imperative programming language Prog to describe programs for hypothesis testing on a dataset that has already sampled from a population. This language has the following two features. First, it has a command for performing a hypothesis test and assigning a test statistic. This command also updates the history of all previously executed hypothesis tests, which is used to calculate the p-values for single and multiple tests. Second, Prog supports parallel compositions of independently running programs to ensure that multiple hypothesis tests do not interfere with each other.

5.1. Syntax of Prog

Let Fsym be the set of all function symbols, where constants are dealt as functions with arity 0. We define the syntax of Prog by the following BNF:

$$T ::= \texttt{bool} \mid \texttt{int} \mid \texttt{real} \mid T \times T \mid \texttt{list}(T) \qquad \text{(Types)}$$

$$e ::= v \mid f(e, \ldots, e) \qquad \text{(Program terms)}$$

$$c ::= \texttt{skip} \mid v := e \qquad \text{(Commands)}$$

$$C ::= c \mid C; C \mid C \parallel C \mid \texttt{if } e \texttt{ then } C \texttt{ else } C \mid \texttt{loop } e \texttt{ do } C \qquad \text{(Programs)}$$

where $v \in \mathsf{Var}_{\mathsf{obs}}$ and $f \in \mathsf{Fsym}$. Then a program can handle only observable variables.

T represents *types*. A type is either `bool` for Boolean values, `int` for integers, `real` for real numbers, $T_1 \times T_2$ for pairs consisting of a value of type T_1 and a value of type T_2, or `list`(T) for lists of values of type T. e represents *expressions* that evaluate to values. An expression is either a variable v or a function call $f(e_1, \ldots, e_k)$; the latter is typically a call to a function that computes a test statistic. c and C represent *commands* and *programs*, respectively. We give their intuitive explanation as follows.

- `skip` does nothing.
- $v := e$ updates v with the result of an evaluation of e.
- $C_1; C_2$ executes C_1 and then C_2.
- $C_1 \parallel C_2$ executes C_1 and C_2 in parallel that may share some data.
- `if` e `then` C_1 `else` C_2 executes C_1 if e evaluates to `true`; executes C_2 if e evaluates to `false`.
- `loop` e `do` C iteratively executes C as long as e evaluates to `true`.

For instance, the programs in Section 9 conform to the programming language Prog.

Hereafter we assume that all programs are well-typed although we do not explicitly mention the types. Checking this condition for our language can be done by adapting a standard type-checking algorithm to our setting.

We write $\mathsf{upd}(C)$ for the set of all variables that may be updated by executing C: $\mathsf{upd}(\texttt{skip}) = \emptyset$, $\mathsf{upd}(v := e) = \{v\}$, $\mathsf{upd}(C_1; C_2) = \mathsf{upd}(C_1 \parallel C_2) = \mathsf{upd}(\texttt{if } e \texttt{ then } C_1 \texttt{ else } C_2) = \mathsf{upd}(C_1) \cup \mathsf{upd}(C_2)$, and $\mathsf{upd}(\texttt{loop } e \texttt{ do } C) = \mathsf{upd}(C)$.

Then we impose the following restriction to every occurrence of $C_1 \parallel C_2$: $\mathsf{upd}(C_1) \cap \mathsf{Var}(C_2) = \mathsf{upd}(C_2) \cap \mathsf{Var}(C_1) = \emptyset$. This restriction is to ensure that an execution of C_1 does not interfere with that of C_2, and vice versa.

$$\langle \texttt{skip}, w\rangle \longrightarrow w;(m, \texttt{skip}, H)$$

$$\langle v := e, w\rangle \longrightarrow w;(m[v \mapsto [\![e]\!]_m], v := e, H)$$

$$\langle v := f_A(y), w\rangle \longrightarrow w;(m', v := f_A(y), H')$$
$$\text{where } m' \stackrel{\text{def}}{=} m[v \mapsto [\![f_A(y)]\!]_m, h_{y,A} \mapsto [\![h_{y,A}+1]\!]_m]$$
$$H' \stackrel{\text{def}}{=} H \uplus \{m(y) \mapsto \{A\}$$

$$\frac{\langle C_1, w\rangle \longrightarrow w'}{\langle C_1; C_2, w\rangle \longrightarrow \langle C_2, w'\rangle} \qquad \frac{\langle C_1, w\rangle \longrightarrow \langle C_1', w'\rangle}{\langle C_1; C_2, w\rangle \longrightarrow \langle C_1'; C_2, w'\rangle}$$

$$\langle \texttt{if } e \texttt{ then } C_1 \texttt{ else } C_2, w\rangle \longrightarrow \begin{cases} \langle C_1, w\rangle & [\![e]\!]_m = \texttt{true} \\ \langle C_2, w\rangle & [\![e]\!]_m = \texttt{false} \end{cases}$$

$$\langle \texttt{loop } e \texttt{ do } C, w\rangle \longrightarrow \begin{cases} \langle C; \texttt{loop } e \texttt{ do } C, w\rangle & [\![e]\!]_m = \texttt{true} \\ w & [\![e]\!]_m = \texttt{false} \end{cases}$$

$$\frac{\langle C_1, w\rangle \longrightarrow \langle C_1', w'\rangle}{\langle C_1 \| C_2, w\rangle \longrightarrow \langle C_1' \| C_2, w'\rangle} \qquad \frac{\langle C_1, w\rangle \longrightarrow w'}{\langle C_1 \| C_2, w\rangle \longrightarrow \langle C_2, w'\rangle}$$

$$\frac{\langle C_2, w\rangle \longrightarrow \langle C_2', w'\rangle}{\langle C_1 \| C_2, w\rangle \longrightarrow \langle C_1 \| C_2', w'\rangle} \qquad \frac{\langle C_2, w\rangle \longrightarrow w'}{\langle C_1 \| C_2, w\rangle \longrightarrow \langle C_1, w'\rangle}$$

Fig. 1. Rules of execution of programs. The operation $\uplus$ on a test history H is defined in (7).

5.2. Semantics of Prog

We define the semantics of Prog over a Kripke model $\mathfrak{M}$ with labeled transitions (Section 4.3) based on the standard structural operational semantics (e.g. [40]). Intuitively, executing a program command c in a possible world $w = (m, a, H)$ updates the memory m, records the command c as the previous action a, and stores all previously executed hypothesis tests in the history H.

Formally, for a possible world $w \in \mathcal{W}$ and $n = \mathsf{len}(w)$, we write

$$w = w[0], w[1], \dots, w[n-2], (m, a, H)$$

where (m, a, H) is the current state $w[n-1]$ with an assignment $m : \mathsf{Var} \to \mathcal{O} \cup \{\bot\}$, an action a in the last transition in w, and a test history H.

For the assignment m of the current state, we define the evaluation $[\![e]\!]_m$ of a program term e inductively by $[\![v]\!]_m = m(v)$ and $[\![f(e_1, \dots, e_k)]\!]_m = [\![f]\!]([\![e_1]\!]_m, \dots, [\![e_k]\!]_m)$.

As in Fig. 1, we define a binary relation

$$\longrightarrow \subseteq (\mathsf{Prog} \times \mathcal{W}) \times ((\mathsf{Prog} \times \mathcal{W}) \cup \mathcal{W})$$

that relates a pair $\langle C, w\rangle$ consisting of a program C and a possible world w to its next step of execution. If C is terminated, the next step will be a possible world w', otherwise the execution continues to the $\langle C', w'\rangle$.

We remark that the semantics of a program contains the trace of commands executed in it. Hence, even if different programs finally result in the same memory, their semantics may be different. For instance, when the value of a variable v is 1, the execution of the two programs $v := v + 1$ and $v := 2 * v$ result in different worlds with the same memory:

$$\langle v := v + 1, ([v \mapsto 1], a, H)\rangle \longrightarrow ([v \mapsto 1], a, H), ([v \mapsto 2], v := v + 1, H)$$

$$\langle v := 2 * v, ([v \mapsto 1], a, H)\rangle \longrightarrow ([v \mapsto 1], a, H), ([v \mapsto 2], v := 2 * v, H)$$

We define the semantic relation $[\![C]\!] \subseteq \mathcal{W} \times \mathcal{W}$ by

$$[\![C]\!](w) = \{w' \mid \langle C, w\rangle \longrightarrow^* w'\}$$

where $\longrightarrow^*$ is the transitive closure of $\longrightarrow$. When the program C does not terminate, $[\![C]\!](w) = \emptyset$. With the semantic relation $[\![c]\!]$ for single commands c of the programming language Prog, we instantiate the transition relation in the Kripke model $\mathfrak{M}$ (Definition 4); i.e., we define the transition relation $\xrightarrow{c}$ as the semantic relation $[\![c]\!]$.

5.2.1. Remark on parallel compositions

Since parallel compositions are nondeterministic, $w' \in [\![C_1 \| C_2]\!](w)$ may not be unique. However, the resulting world w' is essentially the same, because $w' \in [\![C_1 \| C_2]\!](w)$ is convertible to a pair of $w_1 \in [\![C_1]\!](w)$ and $w_2 \in [\![C_1]\!](w)$ and vice versa.

If we have $\langle C_b, w\rangle \longrightarrow^* w; u_b$ for $b = 1, 2$, then by $\mathsf{upd}(C_b) \cap \mathsf{Var}(C_{3-b}) = \emptyset$, we obtain a sequence u' such that $\langle C_1 \| C_2, w\rangle \longrightarrow^* w; u'$ by combining u_1 and u_2.

Conversely, if $\langle C_1 \| C_2, w\rangle \longrightarrow^* w; u'$, we can decompose u' into u_1 and u_2 such that $\langle C_b, w\rangle \longrightarrow^* w; u_b$ for $b = 1, 2$ (for details, see Appendix B.3).

5.2.2. Procedures of hypothesis testing

We define the interpretation of a program f_A for a hypothesis test $A = (\varphi_0, t, D_{t,\varphi_0}, \preccurlyeq^{(s)}, P)$ with a null hypothesis φ_0, a test statistic t, a test type s, and a statistical model P. For a dataset y and an assignment m, $[\![f_A(y)]\!]_m$ represents the p-value:

$$[\![f_A(y)]\!]_m = \Pr_{r \sim D_{t,\varphi_0}} [\, r \preccurlyeq^{(s)} t(m(y))\,], \tag{6}$$

which is the probability that a value r is at most as likely as the test statistic $t(m(y))$ when it is sampled from D_{t,φ_0} in the world where the null hypothesis φ_0 is true.

As in Fig. 1, the execution of a program $v := f_A(y)$ for a hypothesis test A updates the test history H so that A is added to the *multiset* $H(m(y))$ of all tests using the dataset $m(y)$. Formally, the operation $\uplus$ in Fig. 1 is the union of multisets:

$$H'(o) = \begin{cases} H(o) \uplus \{A\} & \text{if } o = m(y) \\ H(o) & \text{otherwise} \end{cases} \tag{7}$$

To refer to the test history H_w in a possible world w, we introduce a *history variable* $h_{y,A} \in \mathsf{Var}_{\mathsf{inv}}$ for each variable y and each hypothesis test A. A history variable $h_{y,A}$ takes an integer value representing the number of executions of a hypothesis test A on a dataset y. Since $h_{y,A}$ is an invisible variable, it never appears in a program.

The interpretation of $h_{y,A}$ is consistent with the test history H_w; namely, $m_w(h_{y,A})$ represents the number of occurrences of A in the multiset $H_w(m_w(y))$. As shown in Fig. 1, if a program command updates H_w, the values $m_w(h_{y,A})$ of history variables $h_{y,A}$ are also updated consistently. Although $h_{y,A}$ is an invisible variable, the test history H_w itself is observable (Definition 3) and is used to define knowledge.

6. Assertion language

In this section, we define an assertion language called *epistemic language for hypothesis testing* (ELHT) that can express knowledge and statistical beliefs.

ELHT is based on the modal logic S5 in which $\mathbf{K}$ is the box modal operator for expressing knowledge. This language has predicates for describing assertions about sampling and hypothesis testing. The statistical belief modality $\mathbf{K}^{<\epsilon}_{y,A}$ is defined as syntax sugar in a form of disjunctive knowledge to express a belief about an alternative hypothesis obtained by a hypothesis test A on a dataset y with a p-value ϵ. Specifically, we formalize a statistical belief $\mathbf{K}^{<\epsilon}_{y,A}\,\varphi$ in an alternative hypothesis φ as the knowledge that either (i) φ holds, (ii) the sampled dataset is unluckily far from the population, or (iii) the population does not satisfy the requirements for the hypothesis test.

6.1. Syntax of the assertion language

We introduce the syntax of the assertion language ELHT. We define assertion terms, formulas, and predicate symbols for statistical notions. Then we introduce the modality of statistical beliefs as disjunctive knowledge.

6.1.1. Assertion terms

We introduce *assertion terms* to denote data values as follows. Recall that Var is the finite set of all variables and Fsym is the set of all function symbols. We introduce a set IntVar of *integer variables* denoting finite tuples of integers such that $\mathsf{IntVar} \cap \mathsf{Var} = \emptyset$. Then the set ATerm of assertion terms is defined by:

$$u ::= x \mid i \mid f(u, \ldots, u)$$

where $x \in \mathsf{Var}$, $i \in \mathsf{IntVar}$, and $f \in \mathsf{Fsym}$. Notice that the assertion terms can deal with invisible variables unlike the program terms (Section 5.1). Fsym includes function symbols denoting families of probability distributions (e.g., N for normal distributions) and those denoting data operations (e.g., mean for calculating the mean of data values).

6.1.2. Assertion formulas

We define the syntax of assertion formulas with a modal operator $\mathbf{K}$ for *knowledge*. As in previous studies, a formula $\mathbf{K}\varphi$ expresses that we know φ. Formally, the set Fml of *formulas* is defined by:

$$\varphi ::= \eta(u_1, \ldots, u_n) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{K}\varphi$$

$$\varphi' ::= \varphi \mid \neg\varphi' \mid \varphi' \wedge \varphi' \mid \forall i.\varphi'$$

where $\eta \in \mathsf{Pred}$, $u_1, \ldots, u_n \in \mathsf{ATerm}$, and $i \in \mathsf{IntVar}$. In the formulas, the quantifiers $\forall$ and $\exists$ never appear inside the epistemic modality $\mathbf{K}$. They are used only to prove the relative completeness of our program logic in later sections. We remark that there is no universal/existential quantification over the observable and invisible variables. We denote the set of all variables occurring in a formula φ by $\mathsf{fv}(\varphi)$.

As syntax sugar, we use *disjunction* $\vee$, *implication* $\to$, and *existential quantifier* $\exists$. We also define *epistemic possibility* $\mathbf{P}$ as usual by $\mathbf{P}\varphi \stackrel{\text{def}}{=} \neg\mathbf{K}\neg\varphi$.

6.1.3. Hypothesis formulas

We introduce notations for alternative/null hypotheses in hypothesis tests (Section 2). Recall that an alternative hypothesis φ_1 is a proposition that we wish to prove, and that a null hypothesis φ_0 is a proposition that contradicts the alternative hypothesis φ_1. We write $\backsim \varphi_1$ for the null hypothesis corresponding to an alternative hypothesis φ_1. In Section 7.1, we define it as syntax sugar and discuss the details.

6.1.4. Predicate symbols

We introduce the following predicate symbols for statistical notions:

- $u = u'$ represents the equality of two data values u and u'.
- $y \overset{n}{\rightsquigarrow} x$ expresses that a dataset y consists of n data sampled from a population x.
- $y \leftharpoonup x$ represents that a dataset y has been sampled from a population x.
- For $\bowtie \in \{=, \leq, \geq, <, >\}$, $\epsilon \in [0,1]$, $y \in \mathsf{Var}$, and $A \in \mathcal{A}$, $\nu^{\bowtie}_{y,A}(\epsilon)$ represents that the observation of a dataset y is unlikely to occur (with exception $\bowtie \epsilon$) according to the hypothesis test A.

For brevity, we define the formula ϖ_S for a multiset S of pairs of variables and hypothesis tests. Intuitively, $\varpi_{(y,A)}$ represents that a dataset y has been sampled from a population that satisfies the hypothesis test A's requirement. Formally:

$$\varpi_S \stackrel{\text{def}}{=} \bigwedge_{(y,A)\in S} y \leftharpoonup P_A(\xi_A, \theta_A),$$

where $P_A(\xi_A, \theta_A)$ denotes the population following the statistical model P_A for a test A (Section 4.4). When S is a singleton $\{(y,A)\}$, we abbreviate $\varpi_{\{(y,A)\}}$ as $\varpi_{y,A}$.

6.1.5. Modality of statistical beliefs

We use the following formulas on executions of hypothesis tests:

- κ describes the record of all hypothesis tests conducted so far. Formally, κ_S is the formula representing that S is the finite multiset of every pair (y,A) of a dataset y and a hypothesis test A that has been applied to y. This formula is defined as equations between history variables $h_{y,A}$ (Section 5.2) and their values by:

$$\kappa_S \stackrel{\text{def}}{=} \bigwedge_{(y,A)\in S} h_{y,A} = n_{(y,A,S)} \wedge \bigwedge_{(y,A)\in(\mathsf{Var}_{\mathsf{obs}}\times\mathcal{A})\setminus S} h_{y,A} = 0 \tag{8}$$

  where $n_{(y,A,S)}$ is the integer representing the number of occurrences of (y,A) in the multiset S, $\mathsf{Var}_{\mathsf{obs}}$ is the finite set of all observable variables, and $\mathcal{A}$ is the finite set of all hypothesis tests we consider. When S is a singleton $\{(y,A)\}$, we abbreviate $\kappa_{\{(y,A)\}}$ as $\kappa_{y,A}$.
- For $\bowtie \in \{=, \leq, \geq, <, >\}$, $\epsilon \in [0,1]$, $y \in \mathsf{Var}$, and a hypothesis test $A = (\varphi_0, t, D_{t,\varphi_0}, \preccurlyeq^{(s)}, P(\xi,\theta))$, we define the formula $\tau^{\bowtie}_{y,A}(\epsilon)$ by:

$$\tau^{\bowtie}_{y,A}(\epsilon) \stackrel{\text{def}}{=} \nu^{\bowtie}_{y,A}(\epsilon) \wedge \kappa_{\ell_{(y,A)}},$$

  where $\ell_{(y,A)}$ is the multiset of each individual test and its dataset in the (combined) hypothesis test A (Equation (5) in Section 4.4).

As syntax sugar, we introduce the *statistical belief modality* $\mathbf{K}^{\bowtie\epsilon}_{y,A}$. Intuitively, a *statistical belief* $\mathbf{K}^{<\epsilon}_{y,A}\,\varphi$ expresses that we believe a hypothesis φ based on a statistical test A on an observed dataset y with a certain error level (p-value) at most ϵ. We formalize this as the knowledge that either (i) the hypothesis φ holds, (ii) the observed dataset y is unluckily far from the population (from which y is sampled), or (iii) the dataset y did not come from a population that satisfies the test A's requirement (e.g., a population following a normal distribution).

Formally, for a hypothesis test $A_{\backsim\varphi}$ with an alternative hypothesis φ and its null hypothesis $\backsim\varphi$, we define:

$$\mathbf{K}^{\bowtie\epsilon}_{y,A_{\backsim\varphi}}\,\varphi \stackrel{\text{def}}{=} \mathbf{K}(\varphi \vee \tau^{\bowtie}_{y,A_{\backsim\varphi}}(\epsilon) \vee \neg\varpi_{y,A_{\backsim\varphi}}). \tag{9}$$

As the dual modality, we define the *statistical possibility* $\mathbf{P}^{\bowtie\epsilon}_{y,A}$ by $\mathbf{P}^{\bowtie\epsilon}_{y,A}\,\varphi \stackrel{\text{def}}{=} \neg\mathbf{K}^{\bowtie\epsilon}_{y,A}\neg\varphi$. For brevity, we often omit the subscript $\backsim\varphi$ from $A_{\backsim\varphi}$ to abbreviate $\mathbf{K}^{\bowtie\epsilon}_{y,A_{\backsim\varphi}}\,\varphi$ as $\mathbf{K}^{\bowtie\epsilon}_{y,A}\,\varphi$. We also write $\mathbf{K}^{\epsilon}_{y,A}$ instead of $\mathbf{K}^{=\epsilon}_{y,A}$ and $\mathbf{K}^{\bowtie\epsilon}\varphi$ instead of $\mathbf{K}^{\bowtie\epsilon}_{y,A}\,\varphi$.

6.2. Semantics of the assertion language

We define semantics for the assertion language ELHT using a Kripke model.

6.2.1. Interpretation of assertion terms and formulas

We introduce an *interpretation function* $\mathcal{I} : \mathsf{IntVar} \to \mathbb{Z}^*$ that assigns a finite tuple of integers to an integer variable. Then we define the interpretation $[\![u]\!]^{\mathcal{I}}_m$ of an assertion term u w.r.t. $\mathcal{I}$ and an assignment $m : \mathsf{Var} \to \mathcal{O} \cup \{\bot\}$ inductively by $[\![x]\!]^{\mathcal{I}}_m = m(x)$, $[\![i]\!]^{\mathcal{I}}_m = \mathcal{I}(i)$, and $[\![f(u_1, \dots, u_k)]\!]^{\mathcal{I}}_m = [\![f]\!]([\![u_1]\!]^{\mathcal{I}}_m, \dots, [\![u_k]\!]^{\mathcal{I}}_m)$.

We define the interpretation of formulas in a world w in a Kripke model $\mathfrak{M} = (\mathcal{W}, (\overset{a}{\hookrightarrow})_{a \in \mathsf{Act}}, \mathcal{R}, (V_w)_{w \in \mathcal{W}})$ in Definition 4 as follows:

$$
\begin{aligned}
\mathfrak{M}, w \vDash^{\mathcal{I}} \eta(u_1, \dots, u_k) \quad &\text{iff} \quad ([\![u_1]\!]^{\mathcal{I}}_{m_w}, \dots, [\![u_k]\!]^{\mathcal{I}}_{m_w}) \in V_w(\eta) \\
\mathfrak{M}, w \vDash^{\mathcal{I}} \neg\varphi \quad &\text{iff} \quad \mathfrak{M}, w \nvDash^{\mathcal{I}} \varphi \\
\mathfrak{M}, w \vDash^{\mathcal{I}} \varphi \wedge \varphi' \quad &\text{iff} \quad \mathfrak{M}, w \vDash^{\mathcal{I}} \varphi \text{ and } \mathfrak{M}, w \vDash^{\mathcal{I}} \varphi' \\
\mathfrak{M}, w \vDash^{\mathcal{I}} \mathbf{K}\varphi \quad &\text{iff} \quad \text{for all } w' \in \mathcal{W},\ (w, w') \in \mathcal{R} \text{ implies } \mathfrak{M}, w' \vDash^{\mathcal{I}} \varphi \\
\mathfrak{M}, w \vDash^{\mathcal{I}} \forall i.\varphi \quad &\text{iff} \quad \mathfrak{M}, w \vDash^{\mathcal{I}[n/i]} \varphi \text{ for all } n \in \mathbb{N}.
\end{aligned}
$$

$\mathfrak{M}$ is sometimes omitted when it is clear from the context.

6.2.2. Interpretation of predicate symbols

We define the interpretation of predicate symbols. Let $A = (\varphi, t, D_{t,\varphi}, \preccurlyeq^{(s)}, P)$ be a hypothesis test. Recall that the population's distribution has type $\mathbb{D}\mathcal{X}$, and that $\preccurlyeq^{(s)}$ is the likeliness relation (Section 4.4). In a world w, we interpret predicate symbols by:

$$
\begin{aligned}
V_w(=) &= \left\{(o, o) \in \mathcal{O} \times \mathcal{O}\right\} \\
V_w(\rightsquigarrow) &= \left\{(d, D, n) \in (\mathsf{list}\,\mathcal{X}) \times (\mathbb{D}\mathcal{X}) \times \mathbb{N} \;\middle|\; \text{There is an } i \in \mathbb{N} \text{ s.t. } w[i] \xrightarrow{d \sim D^n} w[i+1] \right\} \\
V_w(\leftarrowtail) &= \left\{(d, D) \;\middle|\; (d, D, n) \in V_w(\rightsquigarrow)\right\} \\
V_w(\nu^{\bowtie}_{y,A}) &= \left\{\epsilon \in [0,1] \;\middle|\; \Pr_{r \sim D_{t,\varphi}}[\, r \preccurlyeq^{(s)} t(m_w(y))\,] \bowtie \epsilon \right\}.
\end{aligned}
$$

Intuitively, the set $V_w(\nu_{y,A})$ consists of only the p-value ϵ with which the hypothesis test A on the dataset $m_w(y)$ rejects the null hypothesis φ. Then $\mathfrak{M}, w \vDash \nu_{y,A}(\epsilon)$ represents that in a possible world w, the observation of a dataset y is unlikely to occur (except with probability ϵ) according to the hypothesis test A where the test statistic follows the distribution $D_{t,\varphi}$ in the world w.

Formally, we have:

$$
\mathfrak{M}, w \vDash \nu_{y,A}(\epsilon) \;\text{ iff }\; \Pr_{r \sim D_{t,\varphi}}[\, r \preccurlyeq^{(s)} t(m_w(y))\,] = \epsilon, \tag{10}
$$

where the *p-value* $\Pr_{r \sim D_{t,\varphi}}[\, r \preccurlyeq^{(s)} t(m_w(y))\,]$ is the probability that a value r is at most as likely as the test statistic $t(m_w(y))$ when it is sampled from the distribution $D_{t,\varphi}$ in the possible world w; e.g., when $D_{t,\varphi}$ is the standard normal distribution $N(0,1)$, then $\Pr_{r \sim N(0,1)}[\, r \preccurlyeq^{(\mathrm{T})} 1.96\,] = \Pr_{r \sim N(0,1)}[\,|r| \geq 1.96\,] \approx 0.05$. We remark that the p-value is *not* a probability in the real world, but a probability in the possible world w where the null hypothesis φ is true.

Analogously, the interpretation of $\nu^{\bowtie}_{y,A}$ is defined in terms of a range of p-values. For instance, $\nu^{<}_{y,A}(\epsilon)$ represents that the p-value of a test A on a dataset y is less than ϵ.

The interpretation of the formula κ_S is given by:

$$
\mathfrak{M}, w \vDash \kappa_S \;\text{ iff }\; H_w = \biguplus_{(y,A) \in S} \{m_w(y) \mapsto \{A\}\}, \tag{11}
$$

where H_w is the test history that maps a dataset o to the multiset of all hypothesis tests applied to the dataset o in the world w (Section 4.2).

6.3. Interpretation of statistical belief modality

The interpretation of the statistical belief modality $\mathbf{K}^{<\epsilon}_{y,A}$ is given as follows.

$$
\begin{aligned}
\mathfrak{M}, w \vDash \mathbf{K}^{<\epsilon}_{y,A}\varphi \;&\text{ iff }\; \mathfrak{M}, w \vDash \mathbf{K}\big(\varphi \vee \tau^{<}_{y,A}(\epsilon) \vee \neg\varpi_{y,A}\big) \\
&\text{iff for all } w',\ (w, w') \in \mathcal{R} \text{ implies } \mathfrak{M}, w' \vDash (\neg\varphi \wedge \varpi_{y,A}) \to \tau^{<}_{y,A}(\epsilon).
\end{aligned}
$$

Intuitively, $\mathbf{K}^{<\epsilon}_{y,A}\varphi$ expresses a belief that an alternative hypothesis φ on the population is true. For a two-tailed test A, $w' \vDash (\neg\varphi \wedge \varpi_{y,A}) \to \tau^{<}_{y,A}(\epsilon)$ means that if we consider a possible world w' where the null hypothesis $\neg\varphi$ is true and the dataset y is drawn from a population satisfying the test A's requirement $\varpi_{y,A}$, then the execution of A would conclude that the observation of the dataset y is unlikely to occur (with exceptions at most ϵ), i.e., $w' \vDash \tau^{<}_{y,A}(\epsilon)$. See Sections 6.4, 6.5, and 7 for discussion.

Although the modality $\mathbf{K}$ expresses the knowledge in terms of S5, $\mathbf{K}^{<\epsilon}_{y,A}\,\varphi$ represents a *belief* instead of a knowledge. This is because φ can be *false* when $\tau^{<}_{y,A}(\epsilon) \vee \neg\varpi_{y,A}$ holds; i.e., we may have a false belief in φ (i) when the sampled dataset y is unluckily far from the population or (ii) when the dataset y did not come from the population that satisfies the test A's requirement.

Example 4 *(Statistical belief in Z-tests).* Recall again the two-tailed Z-test for two population means in Example 1. The alternative hypothesis is $\varphi \stackrel{\text{def}}{=} (\mu_1 \neq \mu_2)$, and the null hypothesis $\backsim\varphi$ is given by $\mu_1 = \mu_2$. As in Example 3, we denote this Z-test by $A = (\backsim\varphi, t, N(0,1), \preccurlyeq^{(\mathrm{T})}, N(\mu_1,\sigma^2)\times N(\mu_2,\sigma^2))$.

Suppose that in a world w, we sample two datasets $m_w(y_1)$ and $m_w(y_2)$ respectively from two populations $N(\mu_1,\sigma^2)$ and $N(\mu_2,\sigma^2)$. If the null hypothesis $\backsim\varphi$ is true, the Z-test statistic $t(m_w(y_1), m_w(y_2))$ follows the distribution $N(0,1)$.

If $t(m_w(y_1), m_w(y_2)) = 3$, we have:

$$\Pr_{r\sim N(0,1)}[r \preccurlyeq^{(\mathrm{T})} t(m_w(y_1), m_w(y_2))] < 0.05.$$

Then the null hypothesis $\backsim\varphi$ is rejected, and we obtain the statistical belief that the alternative hypothesis φ is true with the significance level 0.05, i.e., $w \vDash \mathbf{K}^{<0.05}_{y,A}\,\varphi$.

In contrast, if $t(m_w(y_1), m_w(y_2)) = 1.8$, then $w \vDash \neg\,\mathbf{K}^{<0.05}_{y,A}\,\varphi$, because we have:

$$\Pr_{r\sim N(0,1)}[\, r \preccurlyeq^{(\mathrm{T})} t(m_w(y_1), m_w(y_2))\,] > 0.05.$$

6.4. Remark on the Universe of the Kripke model

We remark that the universe $\mathcal{W}$ of the model $\mathfrak{M}$ is assumed to include all possible worlds we can imagine. If there is no possible world satisfying a null hypothesis $\backsim\varphi$ in $\mathfrak{M}$, then the alternative hypothesis φ is satisfied in all worlds in $\mathfrak{M}$, hence so are $\mathbf{K}\varphi$ and $\mathbf{K}^{<\epsilon}_{y,A}\,\varphi$. This implies that if we cannot imagine a possible world where $\backsim\varphi$ is true, then we already know that φ is true without conducting the hypothesis test A.

6.5. When hypothesis tests are meaningful

The formula $\mathbf{K}^{<\epsilon}_{y,A}\,\varphi$ expresses a belief after conducting a hypothesis test A on a dataset y, and covers the following two cases where the execution of A is not useful:

(i) we *knew* that the alternative hypothesis φ is true without conducting the test A;
(ii) we know that the requirement $\varpi_{(y,A)}$ for the test A on y is *not* satisfied.

Hence, deriving only the formula $\mathbf{K}^{<\epsilon}_{y,A}\,\varphi$ is not sufficient to conclude the correctness of the alternative hypothesis φ from the execution of the hypothesis test A.

Formally, $\mathbf{K}^{<\epsilon}_{y,A}\,\varphi$ is satisfied also when we have the prior knowledge $\mathbf{K}(\varphi \vee \neg\varpi_{(y,A)})$ that (i) φ is satisfied or (ii) the test A's requirement is not satisfied. Thus, the execution of A is meaningful only when we do *not* have the prior knowledge $\mathbf{K}(\varphi \vee \neg\varpi_{(y,A)})$, i.e., only when we have the *prior belief* $\mathbf{P}(\neg\varphi \wedge \varpi_{(y,A)})$.

For the outcome of the test A to be meaningful, the requirement $\varpi_{(y,A)}$ must hold in the real world. In practice, however, we usually have a limited knowledge of the population (Section 10), and may not know whether the population satisfies the requirement $\varpi_{(y,A)}$. For this reason, in Section 8 and 9, we aim to derive a statistical belief $\mathbf{K}^{<\epsilon}_{y,A}\,\varphi$ under the assumption that $\varpi_{(y,A)}$ holds as a precondition instead of $\mathbf{K}\varpi_{(y,A)}$.

7. Prior beliefs and posterior statistical beliefs in ELHT

In this section, we clarify the importance of prior beliefs in the acquisition of statistical beliefs by describing them using the assertion language ELHT. We then present the essential properties of statistical beliefs; e.g., $\mathbf{K}^{\bowtie\epsilon}_{y,A}$ expresses a belief weaker than the S5 knowledge modality $\mathbf{K}$. We also show how a statistical belief is derived from a test history.

Prior belief/knowledge of hypotheses is essential in choosing which hypothesis testing method is appropriate for a given situation. For example, to apply a two-tailed Z-test, analysts must have the prior belief that both tails ($\mu_1 > \mu_2$ and $\mu_1 < \mu_2$) are possible. In contrast, to apply an upper-tailed test, they must have the prior knowledge that the lower tail ($\mu_1 < \mu_2$) is impossible. Using the assertion language ELHT, we explain that such prior beliefs are important for the application of a hypothesis test to be meaningful.

7.1. Hypothesis formulas

To formalize the prior beliefs for hypothesis testing, we introduce notations for the alternative and null hypotheses in the assertion language ELHT.

We use two formulas φ_{U} and φ_{L} to represent the alternative hypotheses in an upper-tailed test and a lower-tailed test, respectively. Then φ_{U} and φ_{L} cannot be true simultaneously; i.e., $\vDash \neg\varphi_{\mathsf{U}} \vee \neg\varphi_{\mathsf{L}}$. The alternative hypothesis of the two-tailed test is:

Table 2
Hypothesis formulas in the Z-tests (Example 1).

Tails	alternative hypotheses	null hypotheses
Two	$\varphi_{\mathsf{T}} \stackrel{\text{def}}{=} (\mu_1 \neq \mu_2)$	$\backsim \varphi_{\mathsf{T}} \stackrel{\text{def}}{=} (\mu_1 = \mu_2)$
Upper	$\varphi_{\mathsf{U}} \stackrel{\text{def}}{=} (\mu_1 > \mu_2)$	$\backsim \varphi_{\mathsf{U}} \stackrel{\text{def}}{=} (\mu_1 = \mu_2)$
Lower	$\varphi_{\mathsf{L}} \stackrel{\text{def}}{=} (\mu_1 < \mu_2)$	$\backsim \varphi_{\mathsf{L}} \stackrel{\text{def}}{=} (\mu_1 = \mu_2)$

Table 3
Prior belief/knowledge in the Z-tests (Example 1) where φ_{U} and φ_{L} are respectively the alternative hypotheses of the upper-tailed and lower-tailed Z-tests in Table 2.

Tails	prior belief/knowledge	
	general forms	Z-tests
Two	$\mathbf{P}\varphi_{\mathsf{U}} \wedge \mathbf{P}\varphi_{\mathsf{L}}$	$\mathbf{P}(\mu_1 > \mu_2) \wedge \mathbf{P}(\mu_1 < \mu_2)$
Upper	$\mathbf{P}\varphi_{\mathsf{U}} \wedge \neg\mathbf{P}\varphi_{\mathsf{L}}$	$\mathbf{P}(\mu_1 > \mu_2) \wedge \mathbf{K}(\mu_1 \geq \mu_2)$
Lower	$\mathbf{P}\varphi_{\mathsf{L}} \wedge \neg\mathbf{P}\varphi_{\mathsf{U}}$	$\mathbf{P}(\mu_1 < \mu_2) \wedge \mathbf{K}(\mu_1 \leq \mu_2)$

$$\varphi_{\mathsf{T}} \stackrel{\text{def}}{=} \varphi_{\mathsf{U}} \vee \varphi_{\mathsf{L}}. \tag{12}$$

The syntax sugar $\backsim \varphi_{\mathsf{T}}$, $\backsim \varphi_{\mathsf{U}}$, and $\backsim \varphi_{\mathsf{L}}$ for the null hypotheses can be defined by:

$$\backsim \varphi_s \stackrel{\text{def}}{=} \neg\varphi_{\mathsf{U}} \wedge \neg\varphi_{\mathsf{L}} \quad \text{for } s \in \{\mathsf{T}, \mathsf{U}, \mathsf{L}\}. \tag{13}$$

Example 5 *(Hypothesis formulas in Z-tests).* For $\mu_1, \mu_2 \in \mathbb{R}$, the two-tailed, upper-tailed, and lower-tailed Z-test (Example 1) have the alternative hypotheses:

$$\varphi_{\mathsf{T}} \stackrel{\text{def}}{=} (\mu_1 \neq \mu_2), \quad \varphi_{\mathsf{U}} \stackrel{\text{def}}{=} (\mu_1 > \mu_2), \quad \text{and} \quad \varphi_{\mathsf{L}} \stackrel{\text{def}}{=} (\mu_1 < \mu_2).$$

This is because the upper-tailed (resp. lower-tailed) test is based on the assumption $\mu_1 \geq \mu_2$ (resp. $\mu_1 \leq \mu_2$). We can see that $\vDash \varphi_{\mathsf{T}} \leftrightarrow (\varphi_{\mathsf{U}} \vee \varphi_{\mathsf{L}})$ and $\vDash \neg\varphi_{\mathsf{U}} \vee \neg\varphi_{\mathsf{L}}$. The null hypotheses of these tests are $\backsim \varphi_s \stackrel{\text{def}}{=} (\mu_1 = \mu_2)$ for $s \in \{\mathsf{T}, \mathsf{U}, \mathsf{L}\}$. See Table 2 for the summary of the hypothesis formulas in Z-tests.

In the two-tailed test, the null hypothesis $\backsim \varphi_{\mathsf{T}}$ is logically equivalent to $\neg\varphi_{\mathsf{T}}$, i.e., $\mu_1 = \mu_2$. In contrast, in the upper-tailed test, $\backsim \varphi_{\mathsf{U}}$ (i.e., $\mu_1 = \mu_2$) implies $\neg\varphi_{\mathsf{U}}$ (i.e., $\mu_1 \leq \mu_2$) *but not vice versa*. This is also the case with the lower-tailed test.

7.2. Prior beliefs in hypothesis tests

We formally describe the prior knowledge of hypothesis tests using epistemic formulas. We show an example in Table 3.

7.2.1. Prior beliefs in two-tailed Z-tests

For an application of the two-tailed Z-test to be meaningful, we are supposed to have the prior belief that $\mu_1 > \mu_2$ is possible (denoted by $\mathbf{P}\varphi_{\mathsf{U}}$), and that $\mu_1 < \mu_2$ is possible (denoted by $\mathbf{P}\varphi_{\mathsf{L}}$). ELHT naturally explains that these prior beliefs are essential to interpret the results of hypothesis tests as follows. Assume that, in a world w, we had neither of these prior beliefs, but obtained a statistical belief $\mathbf{K}^{\alpha}_{y,A}\,\varphi_{\mathsf{T}}$ by conducting a two-tailed hypothesis test A; i.e., $w \vDash \neg\mathbf{P}\varphi_{\mathsf{U}} \wedge \neg\mathbf{P}\varphi_{\mathsf{L}} \wedge \mathbf{K}^{\alpha}_{y,A}\,\varphi_{\mathsf{T}}$. Since $\mathbf{P}$ is the dual operator of $\mathbf{K}$, we have $w \vDash \mathbf{K}\neg\varphi_{\mathsf{U}} \wedge \mathbf{K}\neg\varphi_{\mathsf{L}} \wedge \mathbf{K}^{\alpha}_{y,A}\,\varphi_{\mathsf{T}}$. By $\varphi_{\mathsf{T}} \stackrel{\text{def}}{=} \varphi_{\mathsf{U}} \vee \varphi_{\mathsf{L}}$, we have:

$$w \vDash \mathbf{K}\neg\varphi_{\mathsf{T}} \wedge \mathbf{K}^{\alpha}_{y,A}\,\varphi_{\mathsf{T}},$$

that is, we already know that the alternative hypothesis φ_{T} is false regardless of the result of the test A (that aims to show that φ_{T} is true). Clearly, the execution of the test A is meaningless when we know that φ_{T} is false. For this reason, the prior beliefs $\mathbf{P}\varphi_{\mathsf{U}}$ and $\mathbf{P}\varphi_{\mathsf{L}}$ are essential for the application of the two-tailed test to be meaningful.

We remark that even if we do not have these prior beliefs, the definition of the formula $\mathbf{K}^{\alpha}_{y,A}\,\varphi_{\mathsf{T}}$ is still consistent with the principle of the hypothesis testing (although the test is useless, as mentioned above). Recall that the statistical belief is defined by $\mathbf{K}^{\alpha}_{y,A}\,\varphi_{\mathsf{T}} \stackrel{\text{def}}{=} \mathbf{K}(\varphi_{\mathsf{T}} \vee \tau_{y,A}(\alpha) \vee \neg\varpi_{y,A})$. Then $w \vDash \mathbf{K}\neg\varphi_{\mathsf{T}} \wedge \mathbf{K}^{\alpha}_{y,A}\,\varphi_{\mathsf{T}}$ implies $w \vDash \mathbf{K}(\tau_{y,A}(\alpha) \vee \neg\varpi_{y,A})$; i.e., we learn that either (i) the sampled dataset y is unluckily far from the population, or (ii) y was sampled from a population that does not satisfy the requirement $\varpi_{y,A}$ for the hypothesis test A on y.

7.2.2. Prior beliefs in one-tailed Z-tests

When we apply the *upper-tailed* Z-test, we are supposed to have the prior belief that $\mu_1 > \mu_2$ is possible (denoted by $\mathbf{P}\varphi_{\mathsf{U}}$), and the prior knowledge that $\mu_1 < \mu_2$ is impossible (denoted by $\neg\mathbf{P}\varphi_{\mathsf{L}}$ or by $\mathbf{K}\neg\varphi_{\mathsf{L}}$).

This prior knowledge $\mathbf{K}\neg\varphi_{\mathsf{L}}$ is used to select an upper-tailed test rather than a two-tailed. In Proposition 1, we show that $\mathbf{K}\neg\varphi_{\mathsf{L}}$ is logically equivalent to $\mathbf{K}(\varphi_{\mathsf{U}} \vee \backsim\varphi_{\mathsf{U}})$; i.e., under the knowledge $\mathbf{K}\neg\varphi_{\mathsf{L}}$, either the alternative hypothesis φ_{U} (i.e., $\mu_1 > \mu_2$) or the null hypothesis $\backsim\varphi_{\mathsf{U}}$ (i.e., $\mu_1 = \mu_2$) holds. Hence, the prior knowledge $\mathbf{K}\neg\varphi_{\mathsf{L}}$ allows for applying the upper-tailed test. Without this prior knowledge, we cannot apply the upper-tailed test, because we do not see whether one of the alternative hypothesis φ_{U} and the null hypothesis $\backsim\varphi_{\mathsf{U}}$ holds.

In conclusion, we can use our assertion logic to explain that the prior knowledge $\mathbf{K}\neg\varphi_{\mathsf{L}}$ is crucial to apply the upper-tailed test. Symmetrically, the lower-tailed test requires the prior knowledge $\mathbf{K}\neg\varphi_{\mathsf{U}}$, as indicated in Table 3.

7.2.3. Posterior beliefs in Z-tests

We remark that the prior beliefs in alternative hypotheses will not change even after conducting hypothesis tests. For example, in the case of the two-tailed Z-test, the prior belief $\mathbf{P}\varphi_{\mathsf{L}} \wedge \mathbf{P}\varphi_{\mathsf{U}}$ remains to hold after conducting the test and obtaining a p-value $\alpha > 0$. This is because the statistical belief is defined as disjunctive knowledge $\mathbf{K}(\varphi_{\mathsf{T}} \vee \tau_{y,A}(\alpha) \vee \neg\varpi_{y,A})$, and thus cannot conclude any knowledge of the alternative hypothesis (e.g., $\mathbf{K}\varphi_{\mathsf{T}}$ or $\mathbf{K}\neg\varphi_{\mathsf{T}}$).

7.2.4. Properties of prior beliefs in hypotheses

Now we show basic properties of prior beliefs in hypotheses as follows.

Proposition 1 *(Basic properties of prior beliefs). Recall that* $\varphi_{\mathsf{T}} \stackrel{\text{def}}{=} \varphi_{\mathsf{U}} \vee \varphi_{\mathsf{L}}$ *and* $\backsim\varphi_s \stackrel{\text{def}}{=} \neg\varphi_{\mathsf{U}} \wedge \neg\varphi_{\mathsf{L}}$ *for each* $s \in \{\mathsf{T}, \mathsf{U}, \mathsf{L}\}$.

1. *In a two-tailed hypothesis test, either the null hypothesis* φ_{T} *or the alternative hypothesis* $\backsim\varphi_{\mathsf{T}}$ *is always satisfied; i.e.,* $\vDash \varphi_{\mathsf{T}} \vee \backsim\varphi_{\mathsf{T}}$.
2. *We know that the lower-tail* φ_{L} *is impossible iff we know that either the null hypothesis* φ_{U} *or the alternative hypothesis* $\backsim\varphi_{\mathsf{U}}$ *for the upper-tail test is satisfied:*

$$\vDash \mathbf{K}\neg\varphi_{\mathsf{L}} \leftrightarrow \mathbf{K}(\varphi_{\mathsf{U}} \vee \backsim\varphi_{\mathsf{U}}).$$

3. *We know that the upper-tail* φ_{U} *is impossible iff we know that either the null hypothesis* φ_{L} *or the alternative hypothesis* $\backsim\varphi_{\mathsf{L}}$ *for the lower-tail test is satisfied:*

$$\vDash \mathbf{K}\neg\varphi_{\mathsf{U}} \leftrightarrow \mathbf{K}(\varphi_{\mathsf{L}} \vee \backsim\varphi_{\mathsf{L}}).$$

The proof is shown in Appendix B.1.

7.3. Type II error

Symmetrically to the p-value (type I error rate) α, the *type II error rate* β is the probability that a hypothesis test A does *not* reject the null hypothesis $\backsim\varphi$ when the alternative hypothesis φ is true. For instance, in the two-tailed Z-test (Example 1), β is the probability that the Z-test fails to reject the null hypothesis $\mu_1 = \mu_2$ when the alternative hypothesis $\mu_1 \neq \mu_2$ is true. We remark that β is determined by the *effect size* $|\mu_1-\mu_2|/\sigma$; For a smaller distance $|\mu_1 - \mu_2|$, it is more difficult for the Z-test to distinguish the null and alternative hypotheses, hence the type II error rate β is larger.

Formally, let y' be a dataset such that the p-value α of a test A is 0.05 in a world w; i.e., $w \vDash \mathbf{K}^{0.05}_{y',A}\,\varphi$. To calculate the type II error rate β, we consider an effect size $es > 0$. Suppose that a hypothesis $\xi \stackrel{\text{def}}{=} (es = |\mu_1-\mu_2|/\sigma)$ is satisfied, i.e., $w \vDash \xi$. The belief about the type II error is expressed by $w \vDash \mathbf{K}^{\beta}_{y',A}\,\neg\xi$; i.e., in the world w, we believe that ξ is false with a degree β of belief, although ξ is actually true in w.

7.4. Properties of statistical beliefs

Next, we present properties of the statistical belief modality $\mathbf{K}^{\bowtie\epsilon}_{y,A}$. Proposition 2 explains basic properties of statistical belief; e.g., $\mathbf{K}^{\bowtie\epsilon}_{y,A}$ expresses a belief weaker than the S5 knowledge $\mathbf{K}$. Proposition 3 shows how a p-value is derived from a test history.

To see these, we remark that the dual operator $\mathbf{P}^{\prec\epsilon}_{y,A}$ represents the statistical possibility; $\mathbf{P}^{\prec\epsilon}_{y,A}\,\varphi$ means that we think a null hypothesis φ may be true after a hypothesis test A did not reject φ with a significance level ϵ. Formally:

$$\begin{aligned} w \vDash \mathbf{P}^{\prec\epsilon}_{y,A}\,\varphi \text{ iff } & \text{there is a } w' \text{ s.t. } (w,w') \in \mathcal{R} \text{ and } w' \nvDash \neg\varphi \vee \tau^{\prec}_{y,A}(\epsilon) \vee \neg\varpi_{y,A} \\ \text{iff } & w \vDash \mathbf{P}(\varphi \wedge \neg\tau^{\prec}_{y,A}(\epsilon) \wedge \varpi_{y,A}). \end{aligned}$$

We obtain the following basic properties of statistical beliefs.

Proposition 2 *(Basic properties of statistical beliefs). Let* $y \in \mathsf{Var}_{\mathsf{obs}}$, $\epsilon, \epsilon' \in \mathbb{R}_{\geq 0}$, *and* $\bowtie \in \{=, \leq, \geq, <, >\}$. *Let* f_A *be a program for a hypothesis test* A *with an alternative hypothesis* φ.

$$\Gamma \vdash \{\psi\}\ \texttt{skip}\ \{\psi\} \qquad \text{(Skip)}$$

$$\frac{\Gamma(v) = \Gamma(e)}{\Gamma \vdash \{\varphi[v \mapsto e]\}\ v := e\ \{\varphi\}} \qquad \text{(UpdVar)}$$

$$\frac{\Gamma \vdash \{\psi\}\ C_1\ \{\psi'\} \qquad \Gamma \vdash \{\psi'\}\ C_2\ \{\varphi\}}{\Gamma \vdash \{\psi\}\ C_1; C_2\ \{\varphi\}} \qquad \text{(Seq)}$$

$$\frac{\Gamma \vdash \{\psi \wedge e\}\ C_1\ \{\varphi\} \quad \Gamma \vdash \{\psi \wedge \neg e\}\ C_2\ \{\varphi\}}{\Gamma \vdash \{\psi\}\ \texttt{if}\ e\ \texttt{then}\ C_1\ \texttt{else}\ C_2\ \{\varphi\}} \qquad \text{(If)}$$

$$\frac{\Gamma \vdash \{\psi \wedge e\}\ C\ \{\psi\}}{\Gamma \vdash \{\psi\}\ \texttt{loop}\ e\ \texttt{do}\ C\ \{\psi \wedge \neg e\}} \qquad \text{(Loop)}$$

$$\frac{\Gamma \models \psi \to \psi' \quad \Gamma \vdash \{\psi'\} C \{\varphi'\} \quad \Gamma \models \varphi' \to \varphi}{\Gamma \vdash \{\psi\} C \{\varphi\}} \qquad \text{(Conseq)}$$

Fig. 2. Axioms and rules for basic constructs for commands.

1. (**SB**ν) *The output of f_A is the p-value of the hypothesis test A on the dataset y; i.e.,* $\models \nu_{y,A}(f_A(y))$.
2. (**SB4**) *If we believe φ based on a test A, then we know this statistical belief; i.e.,* $\models \mathbf{K}^{\bowtie\epsilon}_{y,A}\varphi \to \mathbf{K}\,\mathbf{K}^{\bowtie\epsilon}_{y,A}\varphi$.
3. (**SB5**) *If we failed to reject φ and think it possible, then we know this possibility; i.e.,* $\models \mathbf{P}^{\bowtie\epsilon}_{y,A}\varphi \to \mathbf{K}\,\mathbf{P}^{\bowtie\epsilon}_{y,A}\varphi$.
4. (**SBk**) *Knowledge is also regarded as belief:* $\models \mathbf{K}\varphi \to \mathbf{K}^{\bowtie\epsilon}_{y,A}\varphi$.
5. (**SB-**$<$) *If* $\epsilon \le \epsilon'$, $\models \mathbf{K}^{\epsilon}_{y,A}\varphi \to \mathbf{K}^{\epsilon'}_{y,A}\varphi$ *and* $\models \mathbf{P}^{\epsilon'}_{y,A}\varphi \to \mathbf{P}^{\epsilon}_{y,A}\varphi$.
6. (**SBf**) $\mathbf{K}^{\epsilon}_{y,A}\varphi$ *may be a false belief. The alternative hypothesis φ we believe may be false, i.e., the rejected null hypothesis may be true:* $\epsilon > 0$ *iff* $\not\models \mathbf{K}^{\epsilon}_{y,A}\varphi \to \varphi$.

The proof is shown in Appendix B.1.

We remark that for $\epsilon = 0$, (**SB4**) and (**SB5**) correspond to the axioms (4) and (5) of the modal logic S5, respectively, and thus the statistical belief modality $\mathbf{K}^{0}_{y,A}$ coincides with the knowledge modality $\mathbf{K}$ when the test A's requirement $\varpi_{y,A}$ is satisfied.

Next, we present the relationships between hypothesis tests and statistical beliefs. Recall that a formula of the form κ_S represents a test history. The following proposition allows for deriving a p-value from a test history.

Proposition 3 *(Statistical beliefs by hypothesis tests). Let $y_1, y_2 \in \mathsf{Var}_{\mathsf{obs}}$. Let f_{A_1} and f_{A_2} be programs for hypothesis tests A_1 and A_2 with alternative hypotheses φ_1 and φ_2, respectively. Let $S = \{(y_1, A_1), (y_2, A_2)\}$.*

1. (**BH**κ) *For any $S' \subseteq \mathsf{Var} \times \mathcal{A}$, we have* $\models \kappa_{S'} \leftrightarrow \mathbf{P}\kappa_{S'} \leftrightarrow \mathbf{K}\kappa_{S'}$.
2. (**BHT**) *Let $y \in \mathsf{Var}_{\mathsf{obs}}$ and f_A be a program for a hypothesis tests A with an alternative hypothesis φ. If we execute the test A on the dataset y, then we obtain the statistical belief in φ with the p-value $f_A(y)$; i.e.,* $\models \kappa_{y,A} \to \mathbf{K}^{f_A(y)}_{y,A}\varphi$.
3. (**BHT-**$\vee$) *Let $y \stackrel{\text{def}}{=} (y_1, y_2)$, A be the disjunctive combination of A_1 and A_2, and $\epsilon \stackrel{\text{def}}{=} f_{A_1}(y_1) + f_{A_2}(y_2)$. If we execute A_1 on the dataset y_1 and A_2 on y_2 separately, then we obtain the statistical belief in $\varphi_1 \vee \varphi_2$ with the p-value at most ϵ; i.e.,* $\models \kappa_S \to \mathbf{K}^{\le\epsilon}_{y,A}(\varphi_1 \vee \varphi_2)$.
4. (**BHT-**$\wedge$) *Let $y \stackrel{\text{def}}{=} (y_1, y_2)$, A be the conjunctive combination of A_1 and A_2, and $\epsilon' \stackrel{\text{def}}{=} \min(f_{A_1}(y_1), f_{A_2}(y_2))$. If we execute A_1 on the dataset y_1 and A_2 on y_2 separately, then we obtain the statistical belief in $\varphi_1 \wedge \varphi_2$ with the p-value at most ϵ'; i.e.,* $\models \kappa_S \to \mathbf{K}^{\le\epsilon'}_{y,A}(\varphi_1 \wedge \varphi_2)$.

The proof is shown in Appendix B.1.

Intuitively, (**BH**κ) implies that the analysts know the history S of all previously executed hypothesis tests. Otherwise, they could not check whether a p-value is correctly calculated from the history S. (**BHT**) derives a statistical belief from a history $\{(y, A)\}$ consisting of a single hypothesis test, while (**BHT-**$\vee$) and (**BHT-**$\wedge$) derive statistical beliefs from histories S of multiple hypothesis tests. In Section 8.5, we use these properties to obtain helpful derived rules for belief Hoare logic.

8. Belief Hoare logic for hypothesis testing

We introduce *belief Hoare logic* (BHL) for formalizing and reasoning about statistical inference using hypothesis tests. We define the notions of judgments and partial correctness (Section 8.1) and the inference rules of BHL (Section 8.2). We then show the soundness and relative completeness of BHL (Section 8.3) and present useful derived rules for typical forms of hypothesis testing (Section 8.5).

8.1. Hoare triples

We define an *environment* as a pair $\Gamma = (\Gamma^{\mathsf{inv}}, \Gamma^{\mathsf{obs}})$ consisting of an *invisible environment* Γ^{inv} and an *observable environment* Γ^{obs} that assign types to invisible variables and to observable variables, respectively. We write $\Gamma \models \varphi$ if $\mathfrak{M}, w \models \varphi$ for any model $\mathfrak{M}$ and any

$$\frac{\{h_{y,A} : \mathbb{N}\} \subseteq \Gamma^{\mathsf{inv}},\ \ \{y\colon \mathsf{list}\,\mathcal{X},\ v\colon [0,1]\} \subseteq \Gamma^{\mathsf{obs}},\quad \psi_{\mathsf{pre}} \stackrel{\mathrm{def}}{=} \psi[v \mapsto f_A(y),\ h_{y,A} \mapsto (h_{y,A}+1)]}{(\Gamma^{\mathsf{inv}}, \Gamma^{\mathsf{obs}}) \vdash \{\psi_{\mathsf{pre}}\}\ v := f_A(y)\ \{\psi\}}\ (\textsc{Hist}) \qquad \frac{\Gamma \vdash \{\psi\}\ C_1; C_2\ \{\psi'\}}{\Gamma \vdash \{\psi\}\ C_1 \parallel C_2\ \{\psi'\}}\ (\textsc{Par})$$

Fig. 3. An axiom and a rule for hypothesis tests. (Hist) is the axiom for hypothesis tests. (Par) is the rule for exchanging the sequential composition with the parallel composition.

$$\frac{\{h_{y,A^{(\mathsf{T})}} : \mathbb{N}\} \subseteq \Gamma^{\mathsf{inv}},\ \ \{y\colon \mathsf{list}\,\mathcal{X},\ \alpha\colon [0,1]\} \subseteq \Gamma^{\mathsf{obs}},\quad \alpha, h_{y,A^{(\mathsf{T})}} \notin \mathsf{fv}(\{\varphi_{\mathsf{L}}, \varphi_{\mathsf{U}}, \psi\}),\ \ \Gamma \models \psi \to (\varpi_{y,A^{(\mathsf{T})}} \wedge \mathbf{P}\varphi_{\mathsf{L}} \wedge \mathbf{P}\varphi_{\mathsf{U}})}{(\Gamma^{\mathsf{inv}}, \Gamma^{\mathsf{obs}}) \vdash \{\psi \wedge \kappa_\emptyset\}\ \alpha := f_{A^{(\mathsf{T})}}(y)\ \{\psi \wedge \kappa_{y,A^{(\mathsf{T})}} \wedge \mathbf{K}^{\alpha}_{y,A^{(\mathsf{T})}}\ \varphi_{\mathsf{T}}\}}\ (\textsc{Two-HT})$$

$$\frac{\{h_{y,A^{(\mathsf{L})}} : \mathbb{N}\} \subseteq \Gamma^{\mathsf{inv}},\ \ \{y\colon \mathsf{list}\,\mathcal{X},\ \alpha\colon [0,1]\} \subseteq \Gamma^{\mathsf{obs}},\quad \alpha, h_{y,A^{(\mathsf{L})}} \notin \mathsf{fv}(\{\varphi_{\mathsf{L}}, \varphi_{\mathsf{U}}, \psi\}),\ \ \Gamma \models \psi \to (\varpi_{y,A^{(\mathsf{L})}} \wedge \mathbf{P}\varphi_{\mathsf{L}} \wedge \neg\mathbf{P}\varphi_{\mathsf{U}})}{(\Gamma^{\mathsf{inv}}, \Gamma^{\mathsf{obs}}) \vdash \{\psi \wedge \kappa_\emptyset\}\ \alpha := f_{A^{(\mathsf{L})}}(y)\ \{\psi \wedge \kappa_{y,A^{(\mathsf{L})}} \wedge \mathbf{K}^{\alpha}_{y,A^{(\mathsf{L})}}\ \varphi_{\mathsf{L}}\}}\ (\textsc{Low-HT})$$

$$\frac{\{h_{y,A^{(\mathsf{U})}} : \mathbb{N}\} \subseteq \Gamma^{\mathsf{inv}},\ \ \{y\colon \mathsf{list}\,\mathcal{X},\ \alpha\colon [0,1]\} \subseteq \Gamma^{\mathsf{obs}},\quad \alpha, h_{y,A^{(\mathsf{U})}} \notin \mathsf{fv}(\{\varphi_{\mathsf{L}}, \varphi_{\mathsf{U}}, \psi\}),\ \ \Gamma \models \psi \to (\varpi_{y,A^{(\mathsf{U})}} \wedge \neg\mathbf{P}\varphi_{\mathsf{L}} \wedge \mathbf{P}\varphi_{\mathsf{U}})}{(\Gamma^{\mathsf{inv}}, \Gamma^{\mathsf{obs}}) \vdash \{\psi \wedge \kappa_\emptyset\}\ \alpha := f_{A^{(\mathsf{U})}}(y)\ \{\psi \wedge \kappa_{y,A^{(\mathsf{U})}} \wedge \mathbf{K}^{\alpha}_{y,A^{(\mathsf{U})}}\ \varphi_{\mathsf{U}}\}}\ (\textsc{Up-HT})$$

$$\frac{\{h_{y_2,A_2} : \mathbb{N}\} \subseteq \Gamma^{\mathsf{inv}},\ \ \{y_1 : \mathsf{list}\,\mathcal{X}_1,\ y_2 : \mathsf{list}\,\mathcal{X}_2,\ \alpha_1 : [0,1],\ \alpha_2 : [0,1]\} \subseteq \Gamma^{\mathsf{obs}},\ \ y = (y_1, y_2)\quad \alpha_1, \alpha_2, h_{y_2,A_2} \notin \mathsf{fv}(\{\varphi_1, \varphi_2, \psi\}),\ S = \{(y_1, A_1), (y_2, A_2)\},\ \Gamma \models \psi \to (\varpi_{y_2,A_2} \wedge \mathbf{P}(\varphi_1 \vee \varphi_2))}{(\Gamma^{\mathsf{inv}}, \Gamma^{\mathsf{obs}}) \vdash \{\psi \wedge \kappa_{y_1,A_1} \wedge \mathbf{K}^{\alpha_1}_{y_1,A_1}\varphi_1\}\ \alpha_2 := f_{A_2}(y_2)\ \{\psi \wedge \kappa_S \wedge \mathbf{K}^{\leq \alpha_1+\alpha_2}_{y,A}(\varphi_1 \vee \varphi_2)\}}\ (\textsc{Mult-}\vee)$$

$$\frac{\{h_{y_2,A_2} : \mathbb{N}\} \subseteq \Gamma^{\mathsf{inv}},\ \ \{y_1 : \mathsf{list}\,\mathcal{X}_1,\ y_2 : \mathsf{list}\,\mathcal{X}_2,\ \alpha_1 : [0,1],\ \alpha_2 : [0,1]\} \subseteq \Gamma^{\mathsf{obs}},\ \ y = (y_1, y_2)\quad \alpha_1, \alpha_2, h_{y_2,A_2} \notin \mathsf{fv}(\{\varphi_1, \varphi_2, \psi\}),\ S = \{(y_1, A_1), (y_2, A_2)\},\ \Gamma \models \psi \to (\varpi_{y_2,A_2} \wedge \mathbf{P}(\varphi_1 \wedge \varphi_2))}{(\Gamma^{\mathsf{inv}}, \Gamma^{\mathsf{obs}}) \vdash \{\psi \wedge \kappa_{y_1,A_1} \wedge \mathbf{K}^{\alpha_1}_{y_1,A_1}\varphi_1\}\ \alpha_2 := f_{A_2}(y_2)\ \{\psi \wedge \kappa_S \wedge \mathbf{K}^{\leq \min(\alpha_1,\alpha_2)}_{y,A}(\varphi_1 \wedge \varphi_2)\}}\ (\textsc{Mult-}\wedge)$$

Fig. 4. (Two-HT), (Low-HT), and (Up-HT) are derived rules for a two-tailed test $A^{(\mathsf{T})}$, a lower-tailed $A^{(\mathsf{L})}$, and an upper-tailed $A^{(\mathsf{U})}$, respectively, where φ_{T}, φ_{L}, φ_{U} are alternative hypotheses (Section 7.1) and $\kappa_\emptyset$ is given in (8). (Mult-$\vee$) is for the Bonferroni's method with the disjunctive combination A of two tests A_1 and A_2. (Mult-$\wedge$) is for the conjunctive combination A of A_1 and A_2.

world w that respects the type information in Γ (i.e., the type of $w(v)$ being $\Gamma(v)$ for any $v \in \mathsf{Var}$). Let Env be the set of all possible environments.

A *judgment* is of the form $\Gamma \vdash \{\psi\}\ C\ \{\varphi\}$ where $\Gamma \in \mathsf{Env}$, $\psi, \varphi \in \mathsf{Fml}$, and $C \in \mathsf{Prog}$. Intuitively, this represents that whenever the *precondition* ψ is satisfied, executing the program C results in satisfying the *postcondition* φ if C terminates.

We say that a judgment $\Gamma \vdash \{\psi\}\ C\ \{\varphi\}$ is *valid* iff for any model $\mathfrak{M}$ and any possible world w, if $\mathfrak{M}, w \vDash \psi$, then $\mathfrak{M}, w' \vDash \varphi$ for all $w' \in [\![C]\!](w)$. A valid judgment $\Gamma \vdash \{\psi\}\ C\ \{\varphi\}$ expresses the *partial correctness* of the program C: It respects the precondition ψ and the postcondition φ up to the termination of C.

8.2. Inference rules

We define the inference rules for belief Hoare logic (BHL). The rules consist of those for basic command constructs (Fig. 2) and for hypothesis tests (Fig. 3).

The rules in Fig. 2 for the basic constructs are the same as those for a standard imperative programming language; the readers are referred to a standard textbook on the Hoare logic [5] for details. We add the following remarks to a few rules:

- In the rules (If) and (Loop), the guard condition e is a Boolean expression implicitly used as a logical predicate in the preconditions and the postconditions. Translating a Boolean expression into an ELHT assertion is straightforward.
- The rule (Conseq) refines the precondition and relax the postcondition of a triple. The relation $\Gamma \vDash \varphi$ is used in this rule.

The rules in Fig. 3 are characteristic of BHL. (Hist) describes the properties of an execution of a hypothesis test command f_A on a dataset y. Essentially, this rule states that the precondition is obtained by substituting the p-value $f_A(y)$ for the variable v in the postcondition ψ. (Hist) differs from (UpdVar) in that an execution of f_A on y also increases the history variable $h_{y,A}$ by 1. Recall that $h_{y,A}$ denotes the number of all executions of f_A on y and is updated only by an execution of f_A on y.

The rule (Par) in Fig. 3 exchanges the sequential composition $C_1; C_2$ with the parallel composition $C_1 \parallel C_2$. We recall that in Section 5.1, the restriction $\mathsf{upd}(C_1) \cap \mathsf{Var}(C_2) = \mathsf{upd}(C_2) \cap \mathsf{Var}(C_1) = \emptyset$ is imposed to ensure that an execution of C_1 does not interfere with that of C_2, and vice versa.

8.3. Soundness and relative completeness

We show that BHL satisfies soundness and relative completeness as follows.

Theorem 1 *(Soundness). Every derivable judgment is valid.*

We prove Theorem 1 in Appendix B.4.

In contrast, BHL is *not* complete. As with the standard Hoare logic, the rule (Conseq) uses the validity of assertions $\Gamma \vDash \psi \to \psi'$ and $\Gamma \vDash \varphi' \to \varphi$ as assumptions, which may not have finite proofs because the assertion logic is not complete due to arithmetic.

However, BHL is *relatively complete* [41]: every valid judgment has a finite proof using inference rules of BHL, except for the proofs for the assertions that appear as premises in (Conseq).

Theorem 2 *(Relative completeness). Every valid judgment is derivable except for the proofs for assertions.*

We prove Theorem 2 in Appendix B.5.

8.4. Remarks on decidability

We discuss the decidability of BHL, the assertion logic, and its fragments as follows.

We first remark that BHL is *undecidable*, i.e., there is no effective method for determining whether an arbitrary judgment $\Gamma \vdash \{\varphi\}\ C\ \{\varphi'\}$ is derivable using BHL's inference rules. This undecidability follows from the undecidability of the halting problem; that is, if BHL were decidable, there would be an algorithm that could derive $\Gamma \vdash \{\texttt{true}\}\ C\ \{\texttt{false}\}$ for an arbitrary program C written in the Turing-complete language Prog, i.e., that could determine whether an arbitrary program C terminates or not (hence a contradiction). Nevertheless, undecidable program logic in general is known to be practically useful for real-world programs in many logic-based formal verification techniques, such as [42,43].

Furthermore, our assertion logic (Section 6.1.2) is *undecidable*, since it subsumes the first-order logic with arithmetic. Unlike the first-order logic, even the two-variable, monadic fragment of our assertion logic is also undecidable, because the two-variable, monadic fragment of first-order modal logic is proven to be undecidable when we consider a Kripke frame with a world that is related to infinitely many worlds [44,45].

Finally, we remark that the fragment of our assertion logic without quantifiers over IntVar and without arithmetic is *decidable* due to the decidability of the propositional modal logic S5. In practice, the quantifier-free fragment of our assertion logic can describe the pre-/post-conditions of many popular hypothesis testing methods that involve no loops. As we illustrate in Section 9, our BHL can efficiently reason about practical issues without loops, such as p-value hacking and multiple comparison problems.

8.5. Derived rules

In Fig. 4, we show useful derived rules for typical forms of hypothesis tests. These can be instantiated to a variety of concrete testing methods; see Appendix A.

8.5.1. Derived rules for single hypothesis tests

The derived rules (Two-HT), (Low-HT), and (Up-HT) correspond to two-tailed, lower-tailed, and upper-tailed hypothesis tests, respectively. Recall that the formulas φ_{L}, φ_{U}, and $\varphi_{\mathsf{T}} (\overset{\text{def}}{=} \varphi_{\mathsf{L}} \vee \varphi_{\mathsf{U}})$ denote the alternative hypotheses for the lower-tailed, upper-tailed, and two-tailed tests, respectively (Section 7.1) and $\kappa_{\emptyset}$ is given in (8).

The derived rule (Two-HT) states that we can perform a two-tailed test program $f_{A^{(\mathsf{T})}}$ on a dataset y if we have the prior belief $\mathbf{P}\varphi_{\mathsf{L}} \wedge \mathbf{P}\varphi_{\mathsf{U}}$ that both the lower-tail φ_{L} and upper-tail φ_{U} are possible before performing the test (Section 7.2). If the test $f_{A^{(\mathsf{T})}}$ on y returns a p-value $\alpha \in [0,1]$, we obtain the statistical belief in the alternative hypothesis φ_{T} denoted by $\mathbf{K}^{\alpha}_{y,A^{(\mathsf{T})}} \varphi_{\mathsf{T}}$. The derivation of (Two-HT) is given by:

$$\dfrac{\dfrac{\{h_{y,A^{(\mathsf{T})}} : \mathbb{N}\} \subseteq \Gamma^{\mathsf{inv}},\quad \{y : \mathsf{list}\,\mathcal{X},\ \alpha : [0,1]\} \subseteq \Gamma^{\mathsf{obs}},\quad \alpha, h_{y,A^{(\mathsf{T})}} \notin \mathsf{fv}(\{\varphi_{\mathsf{L}}, \varphi_{\mathsf{U}}, \psi\})}{\Gamma \vdash \{\psi \wedge \kappa_{\emptyset}\}\ \alpha := f_{A^{(\mathsf{T})}}(y)\ \{\psi \wedge \kappa_{y,A^{(\mathsf{T})}}\}}\ \textsc{Hist}}{\Gamma \vdash \{\psi \wedge \kappa_{\emptyset}\}\ \alpha := f_{A^{(\mathsf{T})}}(y)\ \{\psi \wedge \kappa_{y,A^{(\mathsf{T})}} \wedge \mathbf{K}^{\alpha}_{y,A^{(\mathsf{T})}} \varphi_{\mathsf{T}}\}}\ \textsc{Conseq}$$

where (Conseq) uses Proposition 3 (**BHT**). We remark that the derivation does not use $\Gamma \vDash \psi \to (\varpi_{y,A^{(\mathsf{T})}} \wedge \mathbf{P}\varphi_{\mathsf{L}} \wedge \mathbf{P}\varphi_{\mathsf{U}})$. However, for the two-tailed test $A^{(\mathsf{T})}$ to be meaningful, the postcondition must imply $\varpi_{y,A^{(\mathsf{T})}} \wedge \mathbf{P}\varphi_{\mathsf{L}} \wedge \mathbf{P}\varphi_{\mathsf{U}}$, as mentioned in Sections 6.5 and 7.2.

If we have the prior belief $\mathbf{P}\varphi_{\mathsf{L}} \wedge \neg\mathbf{P}\varphi_{\mathsf{U}}$ (resp. $\neg\mathbf{P}\varphi_{\mathsf{L}} \wedge \mathbf{P}\varphi_{\mathsf{U}}$) that only the lower-tail φ_{L} (resp. upper tail φ_{U}) is possible, then we can apply (Low-HT) (resp. (Up-HT)) and obtain the statistical belief $\mathbf{K}^{\alpha}_{y,A^{(\mathsf{L})}} \varphi_{\mathsf{L}}$ (resp. $\mathbf{K}^{\alpha}_{y,A^{(\mathsf{U})}} \varphi_{\mathsf{U}}$). The derivations of (Low-HT) and (Up-HT) are similar to that of (Two-HT).

8.5.2. Derived rules for multiple hypothesis tests

The derived rule (Mult-$\vee$) corresponds to the reasoning about two tests A_1 on y_1 and A_2 on y_2 with a *disjunctive* alternative hypothesis $\varphi_1 \vee \varphi_2$. As illustrated in Section 3, a typical example is to test whether a drug has better efficacy than *at least one* of two drugs.

The precondition in (Mult-∨) expresses that we have obtained a statistical belief $\mathbf{K}^{\alpha_1}_{y_1,A_1}\,\varphi_1$ in an alternative hypothesis φ_1 with a p-value α_1. If we obtain an output α_2 of the second test A_2, we cannot conclude that α_2 is the p-value for φ_2, because the p-value when performing the two tests A_1 and A_2 simultaneously is larger than α_1 and α_2. This is known as the *multiple comparison problem*.

The *Bonferroni's method* is the best-known way to calculate p-values for multiple tests [39]. By applying this method, the p-value in total is bounded above by $\alpha_1+\alpha_2$; i.e., we obtain a statistical belief $\mathbf{K}^{\leq\alpha_1+\alpha_2}_{y,A}(\varphi_1\vee\varphi_2)$ in the alternative hypothesis $\varphi_1\vee\varphi_2$ with the dataset $y\overset{\text{def}}{=}(y_1,y_2)$. In BHL, the derived rule (Mult-∨) guarantees the correct application of the Bonferroni's method; i.e., the inference using BHL does *not* make elementary mistakes (e.g., $\mathbf{K}^{\alpha_2}_{y,A}\,\varphi_2$) where the reported p-value α_2 is lower than the actual p-value in the multiple comparison. The derivation for (Mult-∨) is given by:

$$\dfrac{\dfrac{\alpha_1,\alpha_2,h_{y_2,A_2}\notin \mathsf{fv}(\{\varphi_1,\varphi_2,\psi\}),\ S=\{(y_1,A_1),(y_2,A_2)\}}{\Gamma\vdash\{\psi\wedge\kappa_{y_1,A_1}\}\ \alpha_2:=f_{A_2}(y_2)\ \{\psi\wedge\kappa_S\}}\ \textsc{Hist}}{\Gamma\vdash\{\psi\wedge\kappa_{y_1,A_1}\wedge\mathbf{K}^{\alpha_1}_{y_1,A_1}\varphi_1\}\ \alpha_2:=f_{A_2}(y_2)\ \{\psi\wedge\kappa_S\wedge\mathbf{K}^{\leq\alpha_1+\alpha_2}_{y,A}(\varphi_1\vee\varphi_2)\}}\ \textsc{Conseq}$$

where (Conseq) uses Proposition 3 (**BHT-**∨).

In contrast, the derived rule (Mult-∧) formalizes the reasoning about multiple tests with a *conjunctive* alternative hypothesis $\varphi_1\wedge\varphi_2$ (e.g., the program C_{drug} in Example 2, which tests whether a drug has better efficacy than *both* drugs). According to statistics textbooks (e.g., [39]), this does not make the p-value higher, i.e., the p-value is at most $\min(\alpha_1,\alpha_2)$. (Mult-∧) guarantees the correct procedure for conjunctive hypotheses. The derivation for (Mult-∧) is given by:

$$\dfrac{\dfrac{\alpha_1,\alpha_2,h_{y_2,A_2}\notin \mathsf{fv}(\{\varphi_1,\varphi_2,\psi\}),\ S=\{(y_1,A_1),(y_2,A_2)\}}{\Gamma\vdash\{\psi\wedge\kappa_{y_1,A_1}\}\ \alpha_2:=f_{A_2}(y_2)\ \{\psi\wedge\kappa_S\}}\ \textsc{Hist}}{\Gamma\vdash\{\psi\wedge\kappa_{y_1,A_1}\wedge\mathbf{K}^{\alpha_1}_{y_1,A_1}\varphi_1\}\ \alpha_2:=f_{A_2}(y_2)\ \{\psi\wedge\kappa_S\wedge\mathbf{K}^{\leq\min(\alpha_1,\alpha_2)}_{y,A}(\varphi_1\wedge\varphi_2)\}}\ \textsc{Conseq}$$

where (Conseq) uses Proposition 3 (**BHT-**∧) and $y\overset{\text{def}}{=}(y_1,y_2)$.

9. Reasoning about hypothesis testing procedures using BHL

In this section, we apply our framework to the reasoning about *p-value hacking* and *multiple comparison problems* using BHL.

9.1. Reasoning about p-value hacking

The *p-value hacking* (a.k.a. *data dredging*) is a scientifically malignant technique to obtain a low p-value. A typical example is to conduct hypothesis tests on different datasets and ignore the experiment showing a higher p-value to report only a lower.

Our framework can describe and reason about programs for p-value hacking. For example, the following program C_{hack} conducts a hypothesis test A_1 on a dataset y_1 and another A_2 on y_2, and reports only a lower p-value α while ignoring the higher:

$$C_{\text{hack}}\overset{\text{def}}{=}(\alpha_1:=f_{A_1}(y_1)\parallel\alpha_2:=f_{A_2}(y_2));$$
$$\texttt{if}\ \alpha_1<\alpha_2\ \texttt{then}\ \alpha:=\alpha_1\ \texttt{else}\ \alpha:=\alpha_2.$$

We write φ_1 and φ_2 for the alternative hypotheses of the tests A_1 and A_2, respectively.

Based on the discussion on the prior knowledge in Section 6.5, we assume that we do not have the prior knowledge that these hypotheses are true or the dataset did not come from the population satisfying the requirements of the tests; that is, we have:

$$\psi_{\text{pre}}\overset{\text{def}}{=}\neg\mathbf{K}(\varphi_1\vee\neg\varpi_{y_1,A_1})\wedge\neg\mathbf{K}(\varphi_2\vee\neg\varpi_{y_2,A_2}). \tag{14}$$

For the reported value α to be an actual p-value, the formula

$$\psi^{\text{hack}}_{\text{post}}\overset{\text{def}}{=}\mathbf{K}^{\leq\alpha}_{y_1,A_1}\varphi_1\vee\mathbf{K}^{\leq\alpha}_{y_2,A_2}\varphi_2$$

needs to hold as a postcondition of C_{hack}. Thus, at the end of the first line of C_{hack},

$$\big(\alpha_1<\alpha_2\to(\mathbf{K}^{\leq\alpha_1}_{y_1,A_1}\varphi_1\vee\mathbf{K}^{\leq\alpha_1}_{y_2,A_2}\varphi_2)\big)\wedge\big(\alpha_1\geq\alpha_2\to(\mathbf{K}^{\leq\alpha_2}_{y_1,A_1}\varphi_1\vee\mathbf{K}^{\leq\alpha_2}_{y_2,A_2}\varphi_2)\big)$$

must hold due to the rules (UpdVar) and (If). By applying (Conseq) and the definition of the statistical belief modality, the following formula needs to hold:

$$\mathbf{K}(\varphi_1\vee\tau^{\leq}_{y_1,A_1}(\alpha)\vee\neg\varpi_{y_1,A_1})\vee\mathbf{K}(\varphi_2\vee\tau^{\leq}_{y_2,A_2}(\alpha)\vee\neg\varpi_{y_2,A_2}).$$

By assumption (14), this formula implies $\mathbf{P}\kappa_{y_1,A_1}\vee\mathbf{P}\kappa_{y_2,A_2}$. By Proposition 3 (**BH**κ), we obtain $\kappa_{y_1,A_1}\vee\kappa_{y_2,A_2}$; i.e., only one of the two hypothesis tests has been conducted.

$$\dfrac{\dfrac{\alpha_{12}, \alpha_{13}, h_{y'',A_{13}} \notin \mathsf{fv}(\{\varphi_{12}, \varphi_{13}, \psi\})}{\begin{array}{l}\Gamma \vdash \{\psi_{12}^{\mathsf{post}}\}\ C_{13}\ \{\psi_{13}^{\mathsf{post}}\} \\ \Gamma \models (\psi_{12}^{\mathsf{post}} \wedge \alpha_{12} \leq 0.05) \to \psi_{12}^{\mathsf{post}} \\ \Gamma \models \psi_{13}^{\mathsf{post}} \to \varphi_{\mathsf{post}}\end{array}}\ \textsc{Conseq}}{\Gamma \vdash \left\{\begin{array}{l}\psi_{12}^{\mathsf{post}} \wedge \\ \alpha_{12} \leq 0.05\end{array}\right\}\ C_{13}\ \{\varphi_{\mathsf{post}}\}}\ \textsc{Mult-}\wedge \qquad \dfrac{\dfrac{}{\Gamma \vdash \{\varphi_{\mathsf{post}}\}\ \mathtt{skip}\ \{\varphi_{\mathsf{post}}\}}\ \textsc{Skip}}{\Gamma \models (\psi_{12}^{\mathsf{post}} \wedge \alpha_{12} > 0.05) \to \varphi_{\mathsf{post}}}\ \textsc{Conseq}}{\Gamma \vdash \left\{\begin{array}{l}\psi_{12}^{\mathsf{post}} \wedge \\ \alpha_{12} > 0.05\end{array}\right\}\ \mathtt{skip}\ \{\varphi_{\mathsf{post}}\}}\ \textsc{If}$$

$$\dfrac{\dfrac{\alpha_{12}, h_{y',A_{12}} \notin \mathsf{fv}(\{\varphi_{12}, \psi\})}{\Gamma \vdash \{\psi_{\mathsf{pre}}\}\ C_{12}\ \{\psi_{12}^{\mathsf{post}}\}}\ \textsc{Two-HT} \qquad \Gamma \vdash \{\psi_{12}^{\mathsf{post}}\}\ \begin{array}{l}\mathtt{if}\ \alpha_{12} \leq 0.05 \\ \mathtt{then}\ C_{13} \\ \mathtt{else\ skip}\end{array}\ \{\varphi_{\mathsf{post}}\}}{\Gamma \vdash \{\psi_{\mathsf{pre}}\}\ C_{12};\ \begin{array}{l}\mathtt{if}\ \alpha_{12} \leq 0.05 \\ \mathtt{then}\ C_{13} \\ \mathtt{else\ skip}\end{array}\ \{\varphi_{\mathsf{post}}\}}\ \textsc{Seq}$$

where $\varpi_i \stackrel{\text{def}}{=} y_i \underset{n_i}{\rightsquigarrow} N(\mu_i, \sigma^2)$ $\qquad \alpha \stackrel{\text{def}}{=} \min(\alpha_{12}, \alpha_{13})$

$\psi \stackrel{\text{def}}{=} \bigwedge_{i=1,2,3} \varpi_i \wedge \mathbf{P}(\varphi_{12} \wedge \varphi_{13}) \qquad \psi_{12}^{\mathsf{post}} \stackrel{\text{def}}{=} \psi \wedge \kappa_{y',A_{12}} \wedge \mathbf{K}_{y',A_{12}}^{\alpha_{12}}\ \varphi_{12}$

$\psi_{\mathsf{pre}} \stackrel{\text{def}}{=} \psi \wedge \kappa_{\emptyset} \qquad \psi_{13}^{\mathsf{post}} \stackrel{\text{def}}{=} \psi \wedge \kappa_S \wedge \mathbf{K}_{y,A}^{\leq \alpha}(\varphi_{12} \wedge \varphi_{13})$

$S \stackrel{\text{def}}{=} \{(y', A_{12}), (y'', A_{13})\} \qquad \varphi_{\mathsf{post}} \stackrel{\text{def}}{=} \mathbf{K}_{y',A_{12}}^{\leq 0.05}\ \varphi_{12} \to \mathbf{K}_{y,A}^{\leq \alpha}(\varphi_{12} \wedge \varphi_{13}).$

Fig. 5. An outline of the proof for the illustrating program C_{drug} in Example 2.

$$\dfrac{\dfrac{\alpha_{12},\ h_{y',A_{12}} \notin \mathsf{fv}(\{\varphi_{12}, \psi\})}{\Gamma \vdash \{\psi_{\mathsf{pre}}\}\ C_{12}\ \{\psi_{12}^{\mathsf{post}}\}}\ \textsc{Two-HT} \qquad \dfrac{\dfrac{\alpha_{12},\ \alpha_{13},\ h_{y'',A_{13}} \notin \mathsf{fv}(\{\varphi_{12}, \varphi_{13}, \psi\})}{\Gamma \vdash \{\psi_{12}^{\mathsf{post}}\}\ C_{13}\ \{\psi \wedge \kappa_S \wedge \mathbf{K}_{y,A}^{\leq \alpha_{12}+\alpha_{13}}(\varphi_{12} \vee \varphi_{13})\}}\ \textsc{Mult-}\vee}{\Gamma \vdash \{\psi_{12}^{\mathsf{post}}\}\ C_{13}\ \{\mathbf{K}_{y,A}^{\leq \alpha_{12}+\alpha_{13}}(\varphi_{12} \vee \varphi_{13})\}}\ \textsc{Conseq}}{\Gamma \vdash \{\psi_{\mathsf{pre}}\}\ C_{12};\ C_{13}\ \{\mathbf{K}_{y,A}^{\leq \alpha_{12}+\alpha_{13}}(\varphi_{12} \vee \varphi_{13})\}}\ \textsc{Seq}$$

$$\dfrac{}{\Gamma \vdash \{\psi_{\mathsf{pre}}\}\ C_{12} \parallel C_{13}\ \{\mathbf{K}_{y,A}^{\leq \alpha_{12}+\alpha_{13}}(\varphi_{12} \vee \varphi_{13})\}}\ \textsc{Par}$$

Fig. 6. An outline of the proof for $C_{12} \parallel C_{13}$ where $\psi \stackrel{\text{def}}{=} \bigwedge_{i=1,2,3} \varpi_i \wedge \mathbf{P}(\varphi_{12} \vee \varphi_{13})$, $\psi_{\mathsf{pre}} \stackrel{\text{def}}{=} \psi \wedge \kappa_{\emptyset}$, and $\psi_{12}^{\mathsf{post}} \stackrel{\text{def}}{=} \psi \wedge \kappa_{y',A_{12}} \wedge \mathbf{K}_{y',A_{12}}^{\alpha_{12}}\ \varphi_{12}$.

However, by applying (Par) and (Hist) to C_{hack}'s first line, $\kappa_{\{(y_1,A_1),(y_2,A_2)\}}$ needs to be satisfied; i.e., both the tests must have been conducted. Hence a contradiction. Therefore, we cannot conclude that the reported value α is the actual p-value.

Instead, for $y \stackrel{\text{def}}{=} (y_1, y_2)$ and the disjunctive combination A of A_1 and A_2, we derive that $\mathbf{K}_{y,A}^{\leq \alpha_1+\alpha_2}(\varphi_1 \vee \varphi_2)$ is a postcondition of C_{hack} by using the derived rule (Mult-∨). Therefore, the total p-value $\alpha_1 + \alpha_2$ should be reported without ignoring any experiments.

9.2. Reasoning about multiple comparison with conjunctive alternative hypotheses

We illustrate how BHL reasons about the following program in the multiple comparison in Example 2:

$$C_{\text{drug}} \stackrel{\text{def}}{=} C_{12}; \mathtt{if}\ \alpha_{12} < 0.05\ \mathtt{then}\ C_{13}\ \mathtt{else\ skip},$$

where $C_{12} \stackrel{\text{def}}{=} (\alpha_{12} := f_{A_{12}}(y'))$ is the Z-test A_{12} on $y' = (y_1, y_2)$ with the alternative hypothesis φ_{12}, and $C_{13} \stackrel{\text{def}}{=} (\alpha_{13} := f_{A_{13}}(y''))$ is the Z-test A_{13} on $y'' = (y_1, y_3)$ with φ_{13}. Let A be the conjunctive combination of A_{12} and A_{13}, and $y \stackrel{\text{def}}{=} (y', y'')$.

In this example, the derivation of the judgment $\Gamma \vdash \{\psi_{\mathsf{pre}}\}\ C_{\text{drug}}\ \{\varphi_{\mathsf{post}}\}$ given in (3) guarantees that the hypothesis tests are applied appropriately in the program C_{drug}.

Fig. 5 shows the derivation tree for this judgment. In the derivation, we obtain:

$$\begin{array}{l}\Gamma \vdash \{\psi_{\mathsf{pre}}\}\ C_{12}\ \{\psi_{12}^{\mathsf{post}}\} \\ \Gamma \vdash \{\psi_{12}^{\mathsf{post}} \wedge \alpha_{12} \leq 0.05\}\ C_{13}\ \{\varphi_{\mathsf{post}}\} \\ \Gamma \vdash \{\psi_{12}^{\mathsf{post}} \wedge \alpha_{12} > 0.05\}\ \mathtt{skip}\ \{\varphi_{\mathsf{post}}\}\end{array}$$

where $\psi_{12}^{\mathsf{post}} \stackrel{\text{def}}{=} (\psi \wedge \kappa_{y',A_{12}} \wedge \mathbf{K}_{y',A_{12}}^{\alpha_{12}}\ \varphi_{12})$, $\alpha \stackrel{\text{def}}{=} \min(\alpha_{12}, \alpha_{13})$, and $\varphi_{\mathsf{post}} \stackrel{\text{def}}{=} (\mathbf{K}_{y',A_{12}}^{\leq 0.05}\ \varphi_{12} \to \mathbf{K}_{y,A}^{\leq \alpha}(\varphi_{12} \wedge \varphi_{13}))$. The first judgment is derived using the derived rule (Two-HT). The second judgment is derived by the rules (Mult-∧) and (Conseq). The last judgment is derived from (Skip), (Conseq), and $\Gamma \models (\psi_{12}^{\mathsf{post}} \wedge \alpha_{12} > 0.05) \to \varphi_{\mathsf{post}}$, which is obtained by $\models \mathbf{K}_{y',A_{12}}^{\alpha_{12}}\ \varphi_{12} \wedge \alpha_{12} > 0.05 \to \neg\, \mathbf{K}_{y',A_{12}}^{\leq 0.05}\ \varphi_{12}$. Applying (If) to the last two judgments, we have:

$$\Gamma \vdash \{\psi_{12}^{\mathsf{post}}\}\ \mathtt{if}\ \alpha_{12} \leq 0.05\ \mathtt{then}\ C_{13}\ \mathtt{else\ skip}\ \{\varphi_{\mathsf{post}}\},$$

composing it with the first judgment by applying (Seq), we obtain the judgment in (3).

9.3. Reasoning about multiple comparison with disjunctive alternative hypotheses

In contrast, the program $C_{\text{multi}} \stackrel{\text{def}}{=} C_{12} \,\|\, C_{13}$ in (4) has a disjunctive alternative hypothesis $\varphi_{12} \vee \varphi_{13}$ and thus shows a multiple comparison problem. Fig. 6 show the derivation tree for C_{multi}. Since the alternative hypothesis $\varphi_{12} \vee \varphi_{13}$ is disjunctive, we apply (MULT-$\vee$) to obtain the belief $\mathbf{K}_{y,A}^{\leq \alpha_{12}+\alpha_{13}}(\varphi_{12} \vee \varphi_{13})$, with a p-value (larger than α_{12} and α_{13}) at most $\alpha_{12} + \alpha_{13}$.

10. Discussion

In this section, we provide the whole picture of the justification of statistical beliefs inside and outside BHL. A statistical belief derived in a program relies on the following three issues: (i) the validity of hypothesis testing methods themselves, (ii) the satisfaction of the empirical conditions required for the hypothesis tests, and (iii) the appropriate usage of hypothesis tests in the program. In our framework, these are respectively addressed by (a) the validity of BHL's axioms and rules, (b) the (manual) confirmation of the preconditions in a judgment, and (c) the derivation tree for the judgment.

10.1. Validity of hypothesis testing methods

The validity of hypothesis testing methods is not ensured by mathematics alone. The philosophy of statistics has a long history of argument on the proper interpretation of hypothesis testing. One of the most notable examples is the argument between the frequentist and the Bayesian statistics, which still has many issues to be discussed [46].

We also remark that statistical methods occasionally involve approximation of numerical values. Even when the approximation method has a theoretical guarantee, we may need to confirm the validity of the application of the approximation empirically, e.g., by experiments in the specific situation we apply the statistical methods.

For these reasons, we do not attempt to formalize the "justification" for hypothesis testing methods within BHL, and left them for future work. Instead, we introduce simple (derived) rules that can be instantiated with the hypothesis tests commonly used in practice and explained in textbooks, e.g., [37,38]. Then we focus on the logical aspects of the appropriate usage of hypothesis tests, which has been a long-standing, practical concern but has not been formalized using symbolic logic before.

One of the advantages of this approach is that we do not adhere to a specific philosophy of statistics, but can model both the frequentist and the Bayesian statistics by instantiating the derived rules for hypothesis tests (Appendix A).

10.2. Clarification of empirical conditions

The hypothesis testing methods usually assume some empirical conditions on the unknown population from which the dataset is sampled. Typically, many parametric tests require that the population follows a normal distribution. For instance, the Z-test in Example 1 assumes that the population follows a normal distribution with known variance, but this cannot be rigorously confirmed or justified in general.

In some cases, such conditions on the unknown population are confirmed approximately or partially (i) by exploratory observations of the sampled data and (ii) by prior knowledge of properties of the population (outside the statistical inference). However, there is no general method for justifying such empirical conditions rigorously. Thus, the formal justification of those conditions would require further research in statistics.

In the present paper, the empirical conditions on the unknown population remain to be assumptions from the viewpoint of formal logic. Hence, we describe empirical conditions as the preconditions of a judgment in BHL. Explicit specification of the preconditions would be useful to prevent errors in the choice of statistical methods. Furthermore, when we formalize empirical science in future work, it would be crucial to clarify the empirical conditions that justify scientific conclusions.

10.3. Epistemic aspects of statistical inference

One of our contributions is to show that epistemic logic is useful to formalize statistical inference. Although the outcome of a hypothesis test is the *knowledge* determined by the test action, it may form a false *belief*; i.e., a rejected null hypothesis may be true, and a retained one may be false. Hence, the formalization of statistical inference deals with both truth and beliefs, for which epistemic logic is suitable.

The key to formalizing statistical beliefs is to introduce a Kripke semantics with a possible world where a null hypothesis is true (Section 6.2). This possible world may not be the real world where we actually apply the hypothesis test. Notably, the p-value in the test is the probability defined in this possible world, and not in the real world.

Our Kripke semantics is essential for modeling the appropriate usage of hypothesis tests in the real world. We make a distinction between (i) "ideal" possible worlds where all requirements for the hypothesis tests are satisfied and (ii) the real world where hypothesis tests are actually conducted but their requirements may not be satisfied. Without this distinction, we would deal with only mathematical properties of hypothesis testing methods satisfied in "ideal" possible worlds, and could not discuss the appropriateness of the actual application of the hypothesis tests in the real world.

By using this model, we have clarified that statistical beliefs depend on prior beliefs (Section 7.2). By using the possibility modality **P**, certain requirements for hypothesis tests are formalized as *prior beliefs*, which may not be true or confirmed in the real world.

For example, the choice of two-tailed or one-tailed tests depends on the prior belief that both lower-tail and upper-tail are possible before applying the test.

Finally, the update of statistical beliefs by a hypothesis test is modeled using a transition between possible worlds. Since the world records the history of all hypothesis tests, BHL does not allow for hiding any tests to manipulate the statistics (e.g., in p-value hacking and in multiple comparisons in Section 9).

11. Conclusion

In this work, we proposed a new approach to formalizing and reasoning about statistical inference in programs. Specifically, we introduced belief Hoare logic (BHL) for describing and checking the requirement for applying hypothesis tests appropriately. We proved that this logic is sound and relatively complete w.r.t. the Kripke model for hypothesis tests. Then we showed that BHL is useful for reasoning about practical issues in hypothesis tests. In our framework, we clarified the importance of prior beliefs in acquiring statistical beliefs. We also discussed the whole picture of the justification of statistical inference. We emphasize that this appears to be the first attempt to introduce a program logic for the appropriate application of hypothesis tests.

In ongoing work, we are extending our framework to other kinds of statistical methods [47]. We are also developing a verification tool based on this framework using the same strategy as the existing verifiers based on Hoare logic: (i) synthesizing a proof tree using the proof rules in Fig. 2, (ii) discovering the conditions of the form $\Gamma \vDash \varphi$ that must be valid for the given Hoare triple to hold, and (iii) discharging the discovered conditions using an external solver.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

The authors are supported by ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603), JST. In particular, we thank Ichiro Hasuo for providing the opportunity for us to meet and collaborate in that project. Yusuke Kawamoto is supported by JST, PRESTO Grant Number JPMJPR2022, Japan, and by JSPS KAKENHI Grant Number JP21K12028, Japan. Tetsuya Sato is supported by JSPS KAKENHI Grant Number JP20K19775, Japan. Kohei Suenaga is supported by JST CREST Grant Number JPMJCR2012, Japan. We thank Kenji Fukumizu for providing helpful information on hypothesis testing. We also thank anonymous reviewers and Kentaro Kobayashi for their useful comments on the manuscript.

Appendix A. Instantiation to concrete testing methods

The derived rules for hypothesis tests in Fig. 4 are instantiated with concrete examples of tests given in standard textbooks on statistics (e.g., [37]) as follows.

Example 6 *(Two-tailed Z-test).* We recall the two-tailed Z-test $A_{\varphi_0} = (\varphi_0, t, N(0,1), \lesssim^{(\mathsf{T})}, N(\mu_1,\sigma^2)\times N(\mu_2,\sigma^2))$ in Example 3. By applying the derived rule (Two-HT), the procedure of this test with beliefs is expressed as the valid BHL judgment:

$$\begin{aligned}(\Gamma^{\mathsf{inv}}, \Gamma^{\mathsf{obs}}) \vdash \{&\mathbf{P}(\mu_1 < \mu_2) \wedge \mathbf{P}(\mu_1 > \mu_2) \wedge \kappa_{\emptyset}\}\\ &\alpha := \Pr_{r\sim N(0,1)}\left[\, |r| \geq \left| \frac{\mathsf{mean}(y_1) - \mathsf{mean}(y_2)}{\sigma\sqrt{1/\mathsf{size}(y_1) + 1/\mathsf{size}(y_2)}} \right| \right]\\ &\{\mathbf{K}^{\alpha}_{y,A_{\varphi_0}}(\mu_1 \neq \mu_2) \wedge \mathbf{P}(\mu_1 \neq \mu_2) \wedge \kappa_{y,A_{\varphi_0}}\}.\end{aligned} \tag{A.1}$$

Precisely, the p-value $\Pr_{r\sim N(0,1)}[\ldots]$ in (A.1) is given by the procedure $f_{A_{\varphi_0}}(y)$.

We next show the instantiation to the classical likelihood ratio test with a simple null hypothesis $\xi = \xi_0$ and a simple alternative hypothesis $\xi = \xi_1$ (thus we suppose $\xi_0 \neq \xi_1$), namely, in the setting of the Neyman-Pearson lemma [48].

Example 7 *(Likelihood ratio test).* The goal of the likelihood ratio test is to determine which of two candidate distributions $D_p, D_q \in \mathbb{DR}$ is better to fit a dataset $y = (y_1, \ldots, y_n)$ of sample size n. The alternative hypothesis $\varphi_{\mathsf{L}} \stackrel{\text{def}}{=} (\xi = \xi_1)$ (resp. the null hypothesis $\xi = \xi_0$) represents that the actual distribution is D_p (resp. D_q).

To apply this test, we are expected to have the prior knowledge $\mathbf{K}(\xi = \xi_0 \vee \xi = \xi_1)$ that ξ is either ξ_0 or ξ_1. Let $\varphi_{\mathsf{U}} \stackrel{\text{def}}{=} (\xi \neq \xi_0 \wedge \xi \neq \xi_1)$. Then the prior knowledge is denoted by $\mathbf{K}\neg\varphi_{\mathsf{U}}$, which is logically equivalent to $\neg\mathbf{P}\varphi_{\mathsf{U}}$.

Formally, this test is denoted by $A_{\varphi_0} = (\varphi_0, t, D_{t_\theta,\varphi_0}, \preccurlyeq^{(\mathsf{L})}, P(\xi))$ such that:

$$\varphi_0 \stackrel{\text{def}}{=} (\xi = \xi_0), \quad \varphi_{\mathsf{L}} \stackrel{\text{def}}{=} (\xi = \xi_1), \quad \preccurlyeq^{(\mathsf{L})} \stackrel{\text{def}}{=} \{(r, r') \in \mathbb{R} \times \mathbb{R} \mid r \leq r'\},$$
$$t(y) \stackrel{\text{def}}{=} \frac{\prod_{i=1}^n q(y_i)}{\prod_{i=1}^n p(y_i)}, \quad P(\xi) \stackrel{\text{def}}{=} \begin{cases} D_q & (\xi = \xi_0) \\ D_p & (\xi = \xi_1) \end{cases}, \quad D_{t_\theta,\varphi_0} \stackrel{\text{def}}{=} \frac{\prod_{i=1}^n q(D_q)}{\prod_{i=1}^n p(D_q)}$$

where p and q are the density functions of D_p and D_q, respectively. The probability distributions $p(D_q)$ and $q(D_q)$ are the push-forward measures of D_q along p and q respectively. The likelihood function L is defined by $L(y|\xi_0) = \prod_{i=1}^n q(y_i)$ and $L(y|\xi_1) = \prod_{i=1}^n p(y_i)$, and the test statistic $t(y)$ is called the *likelihood ratio*.

In the likelihood ratio test, for a given p-value α and a threshold k such that $\Pr_{d_1,\dots,d_n \sim D_q}[t((d_1,\dots,d_n)) \leq k] \leq \alpha$, if we have $t(y) \leq k$, the likelihood $L(y|\xi_0)$ is too small to accept the distribution D_q. We then conclude that the other candidate D_p is better to fit y (thus this test is lower-tailed). The p-value of this test is given by:

$$\Pr_{d_1,\dots,d_n \sim D_q}[t((d_1,\dots,d_n)) \leq t(y)]. \tag{A.2}$$

By instantiating the p-value $[\![f_{A_{\varphi_0}}(y)]\!]$, we obtain (A.2). By applying the derived rule (Low-HT), we obtain a valid BHL judgment corresponding the likelihood ratio test:

$$(\Gamma^{\mathsf{inv}}, \Gamma^{\mathsf{obs}}) \vdash \{\mathbf{P}(\xi = \xi_1) \wedge \neg \mathbf{P}(\xi \neq \xi_0 \wedge \xi \neq \xi_1) \wedge \kappa_\emptyset\}$$
$$\alpha := \Pr_{d_1,\dots,d_n \sim D_q}[t((d_1,\dots,d_n)) \leq t(y)]$$
$$\{\mathbf{K}^{\alpha}_{y,A_{\varphi_0}} \wedge \mathbf{P}(\xi = \xi_1) \wedge \kappa_{y,A_{\varphi_0}}\}.$$

We can deal with Bayesian hypothesis tests in an analogous way.

Example 8 *(Bayesian hypothesis test).* Consider the Bayesian likelihood ratio test with a dataset y of sample size n, prior distributions $D_{p'}, D_{q'} \in \mathbb{DR}$ with density functions p' and q', and posterior distributions $D_{p(z)}, D_{q(z)} \in \mathbb{DR}$ with density functions $p(-|z)$ and $q(-|z)$. The goal of this test is to determine whether the dataset y is sampled from $D_{q(z)}$ where z follows $D_{q'}$. The alternative hypothesis $\xi = \xi_1$ (resp. the null hypothesis $\xi = \xi_0$) is that y is sampled from $D_{q(z)}$ where z follows $D_{q'}$ (resp. from $D_{p(z)}$ where z follows $D_{p'}$). As with Example 7, this test requires the prior knowledge $\mathbf{K}(\xi = \xi_0 \vee \xi = \xi_1)$.

We first define the following statistical model with the parameter ξ.

$$(P(\xi))(S) \stackrel{\text{def}}{=} \int_{\mathbb{R}} P_1(\xi, z)(S)\, d\, P_0(\xi)(z) \quad (S \subseteq \mathbb{R} : \text{ measurable})$$
$$\text{where} \quad P_0(\xi) = \begin{cases} D_{q'} & (\xi = \xi_0) \\ D_{p'} & (\xi = \xi_1) \end{cases}, \quad P_1(\xi, z) = \begin{cases} D_{q(z)} & (\xi = \xi_0) \\ D_{p(z)} & (\xi = \xi_1) \end{cases}.$$

In this definition, $P_0(\xi)$ and $P_1(\xi, z)$ are prior and posterior distributions, and $P(\xi)$ is the distribution of y sampled from $P_1(\xi, z)$ where z follows $P_0(\xi)$.

This hypothesis test can be denoted by $A_{\varphi_0} = (\varphi_0, t, D_{t_\theta,\varphi_0}, \preccurlyeq^{(\mathsf{L})}, P(\xi))$ where:

$$\varphi_0 \stackrel{\text{def}}{=} (\xi = \xi_0), \quad \varphi_{\mathsf{L}} \stackrel{\text{def}}{=} (\xi = \xi_1), \quad \preccurlyeq^{(\mathsf{L})} \stackrel{\text{def}}{=} \{(r, r') \in \mathbb{R} \times \mathbb{R} \mid r \leq r'\}$$
$$t(y) \stackrel{\text{def}}{=} \frac{\int q'(z) \prod_{i=1}^n q(y_i|z) dz}{\int p'(z) \prod_{i=1}^n p(y_i|z) dz}, \quad D_{t_\theta,\varphi_0} \stackrel{\text{def}}{=} \frac{\int q'(z) \prod_{i=1}^n q(D_{q(z)}|z) dz}{\int p'(z) \prod_{i=1}^n p(D_{q(z)}|z) dz}.$$

Unlike the (classical) likelihood ratio test, the test statistic $t(y)$ is the *Bayes factor*, that is, the ratio of the marginal likelihoods $L(y|\xi_0) = \int q'(z) \prod_{i=1}^n q(y_i|z) dz$ and $L(y|\xi_1) = \int p'(z) \prod_{i=1}^n p(y_i|z) dz$.

As with the likelihood ratio test, we obtain a valid BHL judgment for the Bayesian hypothesis test by applying the derived rule (Low-HT).

Appendix B. Proofs for technical results

In Appendix B.1, we prove the propositions on statistical beliefs. In Appendix B.2, we show basic results on structural operational semantics. In Appendix B.3, we show remarks on parallel compositions. In Appendix B.4, we prove BHL's soundness. In Appendix B.5, we show BHL's relative completeness. In Table B.4 and B.5, we recall notations used in this paper.

B.1. Proof for properties of statistical beliefs

We show the proof for Proposition 1 as follows.

Table B.4
Notations for syntax.

Symbols	Descriptions
y	Dataset
α, ϵ	p-value
$h_{y,A}$	History variable on a test A on y
f	Function symbol
f_A	Procedure for a test A
C	Program
$y \overset{n}{\leftsquigarrow} x$	y is n data sampled from x
$y \leftsquigarrow x$	y is sampled from x
$\varphi_{\mathsf{T}}, \varphi_{\mathsf{U}}, \varphi_{\mathsf{L}}$	Alternative hypotheses
$\backsim \varphi_{\mathsf{T}}, \backsim \varphi_{\mathsf{U}}, \backsim \varphi_{\mathsf{L}}$	Null hypotheses
$\mathbf{K}^{\bowtie\epsilon}_{y,A}\,\varphi$	Statistical belief on φ

Table B.5
Notations for semantics.

Symbols	Descriptions
$\mathcal{P}(S)$	All multisets over a set S
$\mathbb{D}S$	All distributions over a set S
$\mathfrak{M}$	Kripke model
w	Possible world
m_w	Memory in a world w
H_w	Test history in a world w
$\mathcal{O}$	All data values
a	Action
A	Hypothesis test
$t(y)$	Test statistic of a dataset y
$\preceq_t^{(s)}$	Likeliness relation

Proof. 1. The claim is clear from $\varphi_{\mathsf{T}} \overset{\text{def}}{=} \varphi_{\mathsf{U}} \vee \varphi_{\mathsf{L}}$ and $\backsim \varphi_{\mathsf{T}} \overset{\text{def}}{=} \neg\varphi_{\mathsf{U}} \wedge \neg\varphi_{\mathsf{L}}$.

2. The claim is shown as follows.

$$\begin{aligned} \vDash \mathbf{K}(\varphi_{\mathsf{U}} \vee \backsim \varphi_{\mathsf{U}}) &\Leftrightarrow \vDash \mathbf{K}(\varphi_{\mathsf{U}} \vee (\neg\varphi_{\mathsf{U}} \wedge \neg\varphi_{\mathsf{L}})) && (\text{by } \backsim \varphi_{\mathsf{U}} \overset{\text{def}}{=} \neg\varphi_{\mathsf{U}} \wedge \neg\varphi_{\mathsf{L}}) \\ &\Leftrightarrow \vDash \mathbf{K}(\varphi_{\mathsf{U}} \vee \neg\varphi_{\mathsf{L}}) \\ &\Leftrightarrow \vDash \mathbf{K}\neg\varphi_{\mathsf{L}} && (\text{by } \vDash \varphi_{\mathsf{U}} \to \neg\varphi_{\mathsf{L}}) \end{aligned}$$

3. The proof of this claim is analogous to that of the second claim. □

We show the proof for Proposition 2 as follows.

Proof. 1. (**SB**ν) is straightforward from (10) and (6).

2. We show (**SB4**) as follows. Let w be a world such that $w \vDash \mathbf{K}^{\bowtie\epsilon}_{y,A}\,\varphi$. Then $w \vDash \mathbf{K}(\varphi \vee \tau^{\bowtie}_{y,A}(\epsilon) \vee \neg\varpi_{y,A})$. By the axiom (5) of **K**, we obtain $w \vDash \mathbf{KK}(\varphi \vee \tau^{\bowtie}_{y,A}(\epsilon) \vee \neg\varpi_{y,A})$, hence $w \vDash \mathbf{K}\,\mathbf{K}^{\bowtie\epsilon}_{y,A}\,\varphi$.
3. We show (**SB5**) as follows. Let w be a world such that $w \vDash \mathbf{P}^{\bowtie\epsilon}_{y,A}\,\varphi$. Then $w \vDash \mathbf{P}(\varphi \wedge \neg\tau^{\bowtie}_{y,A}(\epsilon) \wedge \varpi_{y,A})$. By the axiom (4) of **K**, we obtain $w \vDash \mathbf{KP}(\varphi \wedge \neg\tau^{\bowtie}_{y,A}(\epsilon) \wedge \varpi_{y,A})$, and thus $w \vDash \mathbf{K}\,\mathbf{P}^{\bowtie\epsilon}_{y,A}\,\varphi$.
4. We show (**SBk**) as follows. Let w be a world such that $w \vDash \mathbf{K}\varphi$. Since $\mathbf{K}^{\bowtie\epsilon}_{y,A}\,\varphi$ is defined by $\mathbf{K}(\varphi \vee \tau^{\bowtie}_{y,A}(\epsilon) \vee \neg\varpi_{y,A})$, we obtain $w \vDash \mathbf{K}^{\bowtie\epsilon}_{y,A}\,\varphi$. Therefore $\vDash \mathbf{K}\varphi \to \mathbf{K}^{\bowtie\epsilon}_{y,A}\,\varphi$.
5. We show (**SB-**$\prec$) as follows. Recall that $\mathbf{K}^{\epsilon}_{y,A}$ is the abbreviation for $\mathbf{K}^{=\epsilon}_{y,A}$. Assume that $\epsilon \le \epsilon'$. Then we obtain the following formulas on the strength of confidence levels: $\vDash \tau_{y,A}(\epsilon) \to \tau_{y,A}(\epsilon')$. By definition, we obtain $\vDash \mathbf{K}^{\epsilon}_{y,A}\,\varphi \to \mathbf{K}^{\epsilon'}_{y,A}\,\varphi$ and $\vDash \mathbf{P}^{\epsilon'}_{y,A}\,\varphi \to \mathbf{P}^{\epsilon}_{y,A}\,\varphi$.
6. The claim (**SBf**) is immediate from the definition of $\mathbf{K}^{\epsilon}_{y,A}$. □

We show the proof for Proposition 3 as follows.

Proof. 1. We show the direction from left to right in (**BH**κ) as follows. Let w be a world such that $w \vDash \kappa_{S'}$. By $(w, w) \in \mathcal{R}$, we have $w \vDash \mathbf{P}\kappa_{S'}$. Let w' be a world such that $(w, w') \in \mathcal{R}$. Since the test history is observable, $H_{w'} = H_w$; hence $w' \vDash \kappa_{S'}$. Thus, $w \vDash \mathbf{K}\kappa_{S'}$. The other direction can also be shown straightforwardly.

2. We show (**BHT**) as follows. Let w be a world such that $w \vDash \kappa_{y,A}$. Let w' be a world such that $w\mathcal{R}w'$. By definition, we have $obs(w) = obs(w')$, hence $H_w = H_{w'}$. Then, by $w \vDash \kappa_{y,A}$, we obtain $w' \vDash \kappa_{y,A}$. By (**SB**ν), we have $w' \vDash \nu_{y,A}(f_A(y))$. Then by $w' \vDash \kappa_{y,A}$, we obtain $w' \vDash \varphi \vee \tau_{y,A}(f_A(y)) \vee \neg\varpi_{y,A}$. Therefore, we have $w \vDash \mathbf{K}(\varphi \vee \tau_{y,A}(f_A(y)) \vee \neg\varpi_{y,A})$, namely, $w \vDash \mathbf{K}^{f_A(y)}_{y,A}\,\varphi$.

3. We show (**BHT-**$\vee$) as follows. Let $\alpha_1 = f_{A_1}(y_1)$, $\alpha_2 = f_{A_2}(y_2)$, and $\epsilon = \alpha_1 + \alpha_2$. Let w be a world such that $w \models \kappa_S$. Let w' be a world such that $w \mathcal{R} w'$. By definition, we have $obs(w) = obs(w')$, hence $H_w = H_{w'}$. Then, by $w \models \kappa_S$, we obtain $w' \models \kappa_S$. Thus $H_{w'} = \{m_{w'}(y_1) \mapsto \{A_1\}, m_{w'}(y_2) \mapsto \{A_2\}\}$.
Now we show $w' \models \mathbf{K}^{\leq \epsilon}_{(y_1,y_2),A}(\varphi_1 \vee \varphi_2)$ as follows. For $i = 1, 2$, we denote by $A_i = (\backsim \varphi_i, t_i, D_{t_i, \backsim \varphi_i}, \preccurlyeq^{(s_i)}_{t_i}, P_i)$ the hypothesis test with the null hypothesis $\backsim \varphi_i$. For each $i = 1, 2$, by the definition of f_{A_i}, we have the statistical belief that the alternative hypothesis φ_i is true with the significance level α_i; i.e.,

$$\Pr_{r \sim D_{t_i, \backsim \varphi_i}}[\, r \preccurlyeq^{(s_i)}_{t_i} t_i(m_w(y_i))\,] = \alpha_i. \tag{B.1}$$

By $obs(w) = obs(w')$ and $y_1, y_2 \in \mathsf{Var}_{\mathsf{obs}}$,

$$m_{w'}(y_1) = m_w(y_1) \text{ and } m_{w'}(y_2) = m_w(y_2). \tag{B.2}$$

Recall that the disjunctive combination is $A = (\backsim(\varphi_1 \vee \varphi_2), t, D, \preccurlyeq^{(s_1,s_2)}_t, P)$ where $t(y_1, y_2) = (t_1(y_1), t_2(y_2))$, D is a coupling of $D_{t_1, \backsim \varphi_1}$ and $D_{t_2, \backsim \varphi_2}$, and $(r_1, r_2) \preccurlyeq^{(s_1,s_2)}_t (r'_1, r'_2)$ iff either $r_1 \preccurlyeq^{(s_1)}_{t_1} r'_1$ or $r_2 \preccurlyeq^{(s_2)}_{t_2} r'_2$. Then we obtain:

$$\begin{aligned}
&\Pr_{(r_1,r_2)\sim D}[\, (r_1, r_2) \preccurlyeq^{(s_1,s_2)}_t t(m_{w'}(y_1), m_{w'}(y_2))\,] \\
&= \Pr_{(r_1,r_2)\sim D}[\, r_1 \preccurlyeq^{(s_1)}_{t_1} t_1(m_{w'}(y_1)) \vee r_2 \preccurlyeq^{(s_2)}_{t_2} t_2(m_{w'}(y_2))\,] && \text{(by definition)} \\
&\leq \Pr_{r_1 \sim D_{t_1, \backsim \varphi_1}}[\, r_1 \preccurlyeq^{(s_1)}_{t_1} t_1(m_{w'}(y_1))\,] + \Pr_{r_2 \sim D_{t_2, \backsim \varphi_2}}[\, r_2 \preccurlyeq^{(s_2)}_{t_2} t_2(m_{w'}(y_2))\,] \\
&= \alpha_1 + \alpha_2 && \text{(by Equations (B.2), (B.1))} \\
&= \epsilon.
\end{aligned}$$

Recall that $y = (y_1, y_2)$. Then $w' \models \nu^{\leq}_{y,A}(\epsilon)$. Hence by $w' \models \kappa_S$, we obtain $w' \models \varphi \vee \tau^{\leq}_{y,A}(\epsilon)$. Therefore, $w \models \mathbf{K}^{\leq \epsilon}_{y,A}(\varphi_1 \vee \varphi_2)$.

4. We show (**BHT-**$\wedge$) as follows. Let $\alpha_1 = f_{A_1}(y_1)$, $\alpha_2 = f_{A_2}(y_2)$, and $\epsilon' \stackrel{\text{def}}{=} \min(f_{A_1}(y_1), f_{A_2}(y_2))$. Let w be a world such that $w \models \kappa_S$. Let w' be a world such that $w \mathcal{R} w'$. By definition, we have $obs(w) = obs(w')$, hence $H_w = H_{w'}$. Then, by $w \models \kappa_S$, we obtain $w' \models \kappa_S$. Thus $H_{w'} = \{m_{w'}(y_1) \mapsto \{A_1\}, m_{w'}(y_2) \mapsto \{A_2\}\}$.
Now we show $w' \models \mathbf{K}^{\leq \epsilon'}_{(y_1,y_2),A}(\varphi_1 \wedge \varphi_2)$ as follows. For $i = 1, 2$, we denote by $A_i = (\backsim \varphi_i, t_i, D_{t_i, \backsim \varphi_i}, \preccurlyeq^{(s_i)}_{t_i}, P_i)$ the hypothesis test with the null hypothesis $\backsim \varphi_i$. For each $i = 1, 2$, by the definition of f_{A_i}, we have the statistical belief that the alternative hypothesis φ_i is true with the significance level α_i; i.e.,

$$\Pr_{r \sim D_{t_i, \backsim \varphi_i}}[\, r \preccurlyeq^{(s_i)}_{t_i} t_i(m_w(y_i))\,] = \alpha_i. \tag{B.3}$$

By $obs(w) = obs(w')$ and $y_1, y_2 \in \mathsf{Var}_{\mathsf{obs}}$,

$$m_{w'}(y_1) = m_w(y_1) \text{ and } m_{w'}(y_2) = m_w(y_2). \tag{B.4}$$

Recall that the conjunctive combination is $A = (\backsim(\varphi_1 \wedge \varphi_2), t, D, \preccurlyeq^{(s_1,s_2)}_t, P)$ where $t(y_1, y_2) = (t_1(y_1), t_2(y_2))$, D is a coupling of $D_{t_1, \backsim \varphi_1}$ and $D_{t_2, \backsim \varphi_2}$, and $(r_1, r_2) \preccurlyeq^{(s_1,s_2)}_t (r'_1, r'_2)$ iff $r_1 \preccurlyeq^{(s_1)}_{t_1} r'_1$ and $r_2 \preccurlyeq^{(s_2)}_{t_2} r'_2$. Then we obtain:

$$\begin{aligned}
\alpha &\stackrel{\text{def}}{=} \Pr_{(r_1,r_2)\sim D}[\, (r_1, r_2) \preccurlyeq^{(s_1,s_2)}_t t(m_{w'}(y_1), m_{w'}(y_2))\,] \\
&= \Pr_{(r_1,r_2)\sim D}[\, r_1 \preccurlyeq^{(s_1)}_{t_1} t_1(m_{w'}(y_1)) \wedge r_2 \preccurlyeq^{(s_2)}_{t_2} t_2(m_{w'}(y_2))\,] && \text{(by definition)} \\
&\leq \Pr_{(r_1,r_2)\sim D}[\, r_1 \preccurlyeq^{(s_1)}_{t_1} t_1(m_{w'}(y_1))\,] \\
&= \alpha_1 && \text{(by Equations (B.4), (B.3)).}
\end{aligned}$$

A similar inequality holds for α_2. Thus $\alpha \leq \min(\alpha_1, \alpha_2) = \epsilon'$. Recall that $y = (y_1, y_2)$. Then we have $w' \models \nu^{\leq}_{y,A}(\epsilon')$. By $w' \models \kappa_S$, we obtain $w' \models \varphi \vee \tau^{\leq}_{y,A}(\epsilon')$. Therefore, $w' \models \mathbf{K}^{\leq \epsilon'}_{y,A}(\varphi_1 \wedge \varphi_2)$. □

B.2. Basics results on operational semantics

We recall basic results on structural operational semantics. We first show that executions of sequential compositions can be decomposed into ones of its components.

Lemma 1. *Suppose* $\langle C_1; C_2, w\rangle \longrightarrow^k w'$. *There are* $0 < l < k$ *and sequences* u_1, u_2 *such that* $w' = w; u_1; u_2$, $\langle C_1, w\rangle \longrightarrow^l w; u_1$, *and* $\langle C_2, w; u_1\rangle \longrightarrow^{k-l} w; u_1; u_2$.

Proof. We prove by induction on k. If $k = 0, 1$, the statement is vacuously true. If $k = k' + 2$ for $k' \geq 0$, we have one of the following two cases:

$$\langle C_1; C_2, w\rangle \longrightarrow \langle C_1'; C_2, w; u'\rangle \longrightarrow^{k'+1} w', \tag{a}$$

$$\langle C_1; C_2, w\rangle \longrightarrow \langle C_2, w; u'\rangle \longrightarrow^{k'+1} w'. \tag{b}$$

In the case (a), by induction hypothesis, there are $0 < l' < k' + 1$ and sequences u_1', u_2 such that $w' = w; u'; u_1'; u_2$, $\langle C_1', w; u'\rangle \longrightarrow^{l'} w; u'; u_1'$, and $\langle C_2, w; u'; u_1'\rangle \longrightarrow^{k'+1-l'} w'$. Thus, $\langle C_1, w\rangle \longrightarrow^{l'+1} w; (u'; u_1')$ and $\langle C_2, w; (u'; u_1')\rangle \longrightarrow^{k-(l'+1)} w'$. In the case (b), by the definition of execution, we have $\langle C_1, w\rangle \longrightarrow w; u'$ for some u'. □

We recall that the executions of single commands `skip`, $v := e$ and $v := f_A(y)$ are deterministic, hence the semantic relation $[\![a]\!]$ of each action a is functional. These semantic *functions* can be rewritten explicitly as follows:

$$[\![\texttt{skip}]\!](w) = w; (m_w, \texttt{skip}, H_w)$$

$$[\![v := e]\!](w) = w; (m_w[v \mapsto [\![e]\!]_{m_w}], v := e, H_w)$$

$$[\![v := f_A(y)]\!](w) = w; ((m_w[v \mapsto [\![f_A(y)]\!]_{m_w}])\zeta_{h_{y,A}}, v := f_A(y), H_w \uplus \{m_w(y) \mapsto \{A\}\})$$

$$\text{where } (m'\zeta_{h_{y,A}})(v) \stackrel{\text{def}}{=} \begin{cases} m'(h_{y,A}) + 1 & v = h_{y,A} \\ m'(v) & \text{otherwise} \end{cases}$$

We remark here that the *incrementation* $\zeta_{h_{y,A}}$ and the substitution $[v \mapsto [\![f_A(y)]\!]_{m_w}]$ are commutative: $m_w[v \mapsto [\![f_A(y)]\!]_{m_w}])\zeta_{h_{y,A}} = (m_w\zeta_{h_{y,A}})[v \mapsto [\![f_A(y)]\!]_{m_w}])$.

If the memories of the current states of two worlds are identical, execution paths starting at these worlds can be simulated by each other, and can be written explicitly.

Lemma 2. *Let w_1, w_2 be two possible worlds. Suppose that $m_{w_1}(v) = m_{w_2}(v)$ holds for all $v \in \mathsf{Var}(C) \cap \mathsf{Var}_{\mathsf{obs}}$. If $\langle C, w_1\rangle \longrightarrow^k w_1'$ for some w_1', then there are $l \in \mathbb{N}$ with $0 \leq l \leq k$ and a sequence $a_1, a_2, \ldots, a_l$ of actions such that:*

1. *$w_1' = ([\![a_l]\!] \circ \cdots \circ [\![a_1]\!])(w_1)$ holds, and*
2. *$\langle C, w_2\rangle \longrightarrow^k w_2'$ and $m_{w_1'}(v) = m_{w_2'}(v)$ for all $v \in \mathsf{Var}(C) \cap \mathsf{Var}_{\mathsf{obs}}$ hold where $w_2' = ([\![a_l]\!] \circ \cdots \circ [\![a_1]\!])(w_2)$.*

Proof. Suppose $\langle C, w_1\rangle \longrightarrow^k w_1'$. We prove by induction on k. If $k = 0$, the statement holds vacuously. If $k = 1$, we have the following four cases:

- Case $C \equiv \texttt{skip}$. By the definition of $[\![\texttt{skip}]\!]$, we have $l = 1$, $a_1 = \texttt{skip}$ and $w_1' = [\![\texttt{skip}]\!](w_1)$. Let $w_2' = [\![\texttt{skip}]\!](w_2)$. Then we obtain $\langle \texttt{skip}, w_2\rangle \longrightarrow^k w_2'$ and $m_{w_1'}(v) = m_{w_1}(v) = m_{w_2}(v) = m_{w_2'}(v)$ for all $v \in \mathsf{Var}(C) \cap \mathsf{Var}_{\mathsf{obs}}$.
- Case $C \equiv (v := e)$. By the definition of $[\![v := e]\!]$, we have $l = 1$, $a_1 = (v := e)$ and $w_1' = [\![v := e]\!](w_1)$. Let $w_2' = [\![v := e]\!](w_2)$. Then we obtain $\langle v := e, w_2\rangle \longrightarrow^k w_2'$ and for all $v' \in \mathsf{Var}(C) \cap \mathsf{Var}_{\mathsf{obs}}$,

$$m_{w_1'}(v') = m_{w_1}[v \mapsto [\![e]\!]_{m_{w_1}}](v') = m_{w_2}[v \mapsto [\![e]\!]_{m_{w_2}}](v') = m_{w_2'}(v').$$

- Case $C \equiv (v := f_A(y))$. By the definition of $[\![v := f_A(y)]\!]$, we have $l = 1$, $a_1 = (v := f_A(y))$ and $w_1' = [\![v := f_A(y)]\!](w_1)$. Let $w_2' = [\![v := f_A(y)]\!](w_2)$. Then we obtain $\langle v := e, w_2\rangle \longrightarrow^k w_2'$ and for all $v' \in \mathsf{Var}(C) \cap \mathsf{Var}_{\mathsf{obs}}$,

$$m_{w_1'}(v') = m_{w_1}[v \mapsto [\![f_A(y)]\!]_{m_{w_1}}](v') = m_{w_2}[v \mapsto [\![f_A(y)]\!]_{m_{w_2}}](v') = m_{w_2'}(v').$$

- Case $C \equiv \texttt{loop}\ e\ \texttt{do}\ C'$ and $[\![e]\!]_{m_w} = \texttt{false}$. We have $l = 0$ and for all $v \in \mathsf{Var}(C) \cap \mathsf{Var}_{\mathsf{obs}}$, $m_{w_1'}(v) = m_{w_1}(v) = m_{w_2}(v) = m_{w_2'}(v)$.

If $k = k' + 2$ for $k' \geq 0$, we have $\langle C, w_1\rangle \longrightarrow \langle C', w_1''\rangle \longrightarrow^{k'+1} w_1'$. By induction hypothesis, there is $a_1, \ldots, a_{l'}$ such that $w_1' = ([\![a_{l'}]\!] \circ \cdots \circ [\![a_1]\!])(w_1'')$ for some $l' \leq k' + 1$, and if $m_{w_1''}(v) = m_{w_2''}(v)$ holds for all $v \in \mathsf{Var}(C') \cap \mathsf{Var}_{\mathsf{obs}}$ then $\langle C', w_2''\rangle \longrightarrow^{k'+1} w_2'$ and $m_{w_1'}(v) = m_{w_2'}(v)$ holds for all $v \in \mathsf{Var}(C') \cap \mathsf{Var}_{\mathsf{obs}}$ where $w_2' = ([\![a_{l'}]\!] \circ \cdots \circ [\![a_1]\!])(w_2'')$.

It suffices to show that for the first step $\langle C, w_1\rangle \longrightarrow \langle C', w_1''\rangle$ of execution,

1. at most a single action is performed, i.e., either $w_1'' = w_1$ or $w_1'' = [\![a']\!](w_1)$, and
2. if $m_{w_1}(v) = m_{w_2}(v)$ for all $v \in \mathsf{Var}(C) \cap \mathsf{Var}_{\mathsf{obs}}$, then $\langle C, w_2\rangle \longrightarrow \langle C', w_2''\rangle$ where $w_2'' = w_2$ if $w_1'' = w_1$, and $w_2'' = [\![a']\!](w_2)$ if $w_1'' = [\![a']\!](w_1)$.

We prove this by induction on the inference tree as follows. Recall that by assumption, $m_{w_1}(v) = m_{w_2}(v)$ holds for all $v \in \mathsf{Var}(C) \cap \mathsf{Var}_{\mathsf{obs}}$.

- If the inference is $\langle \texttt{loop}\ e\ \texttt{do}\ C_1, w_1\rangle \longrightarrow \langle C_1; \texttt{loop}\ e\ \texttt{do}\ C_1, w_1''\rangle$ where $[\![e]\!]_{m_{w_1}} = \texttt{true}$, then $w_1'' = w_1$. From $[\![e]\!]_{m_{w_2}} = [\![e]\!]_{m_{w_1}} = \texttt{true}$, we conclude $\langle \texttt{loop}\ e\ \texttt{do}\ C_1, w_2\rangle \longrightarrow \langle C_1; \texttt{loop}\ e\ \texttt{do}\ C_1, w_2''\rangle$ where $w_2'' = w_2$.
- Similarly, if the inference is $\langle \texttt{if}\ e\ \texttt{then}\ C_1 \texttt{else}\ C_2, w_1\rangle \longrightarrow \langle C_1, w_1''\rangle$, where $[\![e]\!]_{m_{w_1}} = \texttt{true}$, then $w_1'' = w_1$ and $\langle \texttt{if}\ e\ \texttt{then}\ C_1\ \texttt{else}\ C_2, w_2\rangle \longrightarrow \langle C_1, w_2\rangle$.
- If the last step of inference is derived by one of the following rules

$$\frac{\langle C_1, w\rangle \longrightarrow \langle C_1', w''\rangle}{\langle C_1; C_2, w\rangle \longrightarrow \langle C_1'; C_2, w''\rangle}, \quad \frac{\langle C_1, w\rangle \longrightarrow \langle C_1', w''\rangle}{\langle C_1 \| C_2, w\rangle \longrightarrow \langle C_1' \| C_2, w''\rangle}$$

  then we apply the induction hypothesis to $\langle C_1, w_1\rangle \longrightarrow \langle C_1', w_1''\rangle$. We have either $w_1'' = w_1$ or $w_1'' = [\![a']\!](w_1)$, and $\langle C_1, w_2\rangle \longrightarrow \langle C_1', w_2''\rangle$ where $w_2'' = w_2$ if $w_1'' = w_1$, and $w_2'' = [\![a']\!](w_2)$ if $w_1'' = [\![a']\!](w_1)$. Then by the same rule, we conclude $\langle C, w_2\rangle \longrightarrow \langle C', w_2''\rangle$.
- The other cases are shown in a similar way. □

B.3. Remarks on parallel compositions

We present some remarks on parallel compositions. We first show that in general, parallel compositions contain sequential compositions.

Lemma 3. *For any possible world w, we have $[\![C_1; C_2]\!](w) \subseteq [\![C_1 \| C_2]\!](w)$.*

Proof. Suppose $\langle C_1; C_2, w\rangle \longrightarrow^* w'$ Thanks to Lemma 1, there are $l, k > 0$ and $w'' \in \mathcal{W}$ such that $\langle C_1, w\rangle \longrightarrow^l w''$ and $\langle C_2, w''\rangle \longrightarrow^k w'$. We show $\langle C_1 \| C_2, w\rangle \longrightarrow^{l+k} w'$ by induction on l. If $l = 0$, the statement holds vacuously. If $l = l' + 1$, the inference $\langle C_1, w\rangle \longrightarrow^l w''$ can be decomposed into $\langle C_1, w\rangle \longrightarrow \langle C_1', w_0\rangle \longrightarrow^{l'} w''$ for some $w_0 \in \mathcal{W}$. Hence, $\langle C_1'; C_2, w_0\rangle \longrightarrow^{l'} \langle C_2, w''\rangle \longrightarrow^k w'$. By induction hypothesis, we obtain $\langle C_1 \| C_2, w\rangle \longrightarrow \langle C_1' \| C_2, w_0\rangle \longrightarrow^{l'+k} w'$. This completes the proof. □

Next, we show that for a world w and a parallel composition $C_1 \| C_2$, a world $w' \in [\![C_1 \| C_2]\!](w)$ is convertible to a pair of $w_1 \in [\![C_1]\!](w)$ and $w_2 \in [\![C_2]\!](w)$ and vice versa. Recall that we imposed the restriction $\mathsf{upd}(C_b) \cap \mathsf{Var}(C_{3-b}) = \emptyset$ for $b = 1, 2$.

Let us consider $\langle C_b, w\rangle \longrightarrow^* w_b'$ for $b = 1, 2$. By Lemma 2, we obtain $w_b' = ([\![a_{l_b}^b]\!] \circ \cdots \circ [\![a_1^b]\!])(w)$ for $b = 1, 2$, $l_b \geq 0$, and a sequence $a_1^b, \ldots, a_{l_b}^b$ of actions. Then, we can define the following possible world:

$$w' = ([\![a_{l_2}^2]\!] \circ \cdots \circ [\![a_1^2]\!] \circ [\![a_{l_1}^1]\!] \circ \cdots \circ [\![a_1^1]\!])(w).$$

We first show that this can be an execution of $C_1 \| C_2$ starting at the world w.

Lemma 4. *If $\langle C_b, w\rangle \longrightarrow^* ([\![a_{l_b}^b]\!] \circ \cdots \circ [\![a_1^b]\!])(w)$ holds for each $b = 1, 2$, then we have $\langle C_1 \| C_2, w\rangle \longrightarrow^* ([\![a_{l_2}^2]\!] \circ \cdots \circ [\![a_1^2]\!] \circ [\![a_{l_1}^1]\!] \circ \cdots \circ [\![a_1^1]\!])(w)$.*

Proof. Let $w_3 = ([\![a_{l_1}^1]\!] \circ \cdots \circ [\![a_1^1]\!])(w)$. By the assumptions of this lemma, we have $\langle C_1 \| C_2, w\rangle \longrightarrow^* \langle C_2, w_3\rangle$. Since $\mathsf{upd}(C_1) \cap \mathsf{Var}(C_2) = \emptyset$, $m_{w_3}(v) = m_w(v)$ holds for all $v \in \mathsf{Var}(C_2) \cap \mathsf{Var}_{\mathsf{obs}}$. By Lemma 2, we conclude:

$$\langle C_1 \| C_2, w\rangle \longrightarrow^* \langle C_2, w_3\rangle \longrightarrow^* ([\![a_{l_2}^2]\!] \circ \cdots \circ [\![a_1^2]\!] \circ [\![a_{l_1}^1]\!] \circ \cdots \circ [\![a_1^1]\!])(w). \quad \square$$

Second, we show the converse of the above lemma. Let w' be a world such that $\langle C_1 \| C_2, w\rangle \longrightarrow^* w'$. By Lemma 2, there is a sequence $a_1, \ldots, a_{n'}$ of actions such that:

$$w' = ([\![a_{n'}]\!] \circ \cdots \circ [\![a_1]\!])(w).$$

Then we can decompose it into executions of C_1 and C_2 in the following sense.

Lemma 5. *The sequence $a_1, \ldots, a_{n'}$ of actions can be decomposed into two subsequences $a_{L_1^1}, \ldots, a_{L_{n_1'}^1}$ and $a_{L_1^2}, \ldots, a_{L_{n_2'}^2}$ such that for each $b = 1, 2$, $\langle C_b, w\rangle \longrightarrow^* w_b'$ and $w_b' = ([\![a_{L_{n_b'}^b}]\!] \circ \cdots \circ [\![a_{L_1^b}]\!])(w)$.*

Proof. By assumption, there is a $k \geq 0$ such that $\langle C_1 \| C_2, w\rangle \longrightarrow^k w'$. We prove this lemma by induction on k. If $k = 0, 1$, the statement holds vacuously. Suppose $k = k' + 2$ for $k' \geq 0$. We decompose that execution into $\langle C_1 \| C_2, w\rangle \longrightarrow \gamma \longrightarrow^{k'+1} w'$.

- Case $\gamma \equiv \langle C_1' \| C_2, w''\rangle$ for some C_1' and w''. By definition, we should have $\langle C_1, w\rangle \longrightarrow \langle C_1', w''\rangle$. Then we have the following two cases.
  - Case $w'' = w$. By induction hypothesis, we obtain $\langle C_1', w\rangle \longrightarrow^* w_1'$ and $\langle C_2, w\rangle \longrightarrow^* w_2'$. Then, we also have $\langle C_1, w\rangle \longrightarrow \langle C_1', w\rangle \longrightarrow^* w_1'$.

– Case $w'' = [\![a_1]\!](w)$. We have $L^1_1 = 1$. By applying the induction hypothesis to $\langle C'_1 \| C_2, [\![a_1]\!](w)\rangle \longrightarrow^{k'+1} w'$, there are two subsequences $a_{L^1_2}, \ldots, a_{L^1_{n'_1}}$ and $a_{L^2_1}, \ldots, a_{L^2_{n'_1}}$ of $a_2, \ldots a_{n'}$ such that:

$$\langle C'_1, [\![a_1]\!](w)\rangle \longrightarrow^* ([\![a_{L^1_{n'_1}}]\!] \circ \cdots \circ [\![a_{L^1_1}]\!])([\![a_1]\!](w)) = w'_1,$$

$$\langle C_2, [\![a_1]\!](w)\rangle \longrightarrow^* ([\![a_{L^2_{n'_2}}]\!] \circ \cdots \circ [\![a_{L^2_1}]\!])([\![a_1]\!](w)).$$

Since the action a_1 is performed in the program C_1 and $\mathsf{upd}(C_1) \cap \mathsf{Var}(C_2) = \emptyset$, we have $m_{[\![a_1]\!](w)}(v) = m_w(v)$ for all $v \in \mathsf{Var}(C_2) \cap \mathsf{Var}_{\mathsf{obs}}$. Thus, by Lemma 2, we conclude:

$$\langle C_2, w\rangle \longrightarrow^* ([\![a_{L^2_{n'_2}}]\!] \circ \cdots \circ [\![a_{L^2_1}]\!])(w) = w'_2.$$

- Case $\gamma \equiv \langle C_2, w''\rangle$ for some w''. By definition, we should have $\langle C_1, w\rangle \longrightarrow w''$ and $\langle C_2, w''\rangle \longrightarrow^* w'$. Then $w'' = w'_1$ and $m_w(v) = m_{w'_1}(v)$ holds for all $v \in \mathsf{Var}(C_2) \cap \mathsf{Var}_{\mathsf{obs}}$. We have the following two cases.
  – Case $w'_1 = w$. We immediately obtain $\langle C_1, w\rangle \longrightarrow w$ and $\langle C_2, w\rangle \longrightarrow^* w' = ([\![a_{n'}]\!] \circ \cdots \circ [\![a_1]\!])(w)$.
  – Case $w_1 = [\![a_1]\!](w)$. We have $L^1_1 = 1$, $\langle C_1, w\rangle \longrightarrow [\![a_1]\!](w)$, and $\langle C_2, [\![a_1]\!](w)\rangle \longrightarrow^* w'$. Since a_1 belongs to executions in C_1 and $\mathsf{upd}(C_1) \cap \mathsf{Var}(C_2) = \emptyset$, we have $m_{[\![a_1]\!](w)}(v) = m_w(v)$ for all $v \in \mathsf{Var}(C_2) \cap \mathsf{Var}_{\mathsf{obs}}$. Thus by Lemma 2, we obtain $\langle C_2, w\rangle \longrightarrow^* ([\![a_{n'}]\!] \circ \cdots \circ [\![a_2]\!])(w)$.
- The other cases are proved in a similar way. □

Executions of programs can be nondeterministic due to parallel compositions. However, since two programs in parallel do not interfere with each other, their executions result in the same memory and test history as follows.

Lemma 6. *For any $w' \in [\![C_1 \| C_2]\!](w)$, there is a $w^* \in [\![C_1; C_2]\!](w)$ such that $m_{w'} = m_{w^*}$ and $H_{w'} = H_{w^*}$.*

Proof. Let $w' \in [\![C_1 \| C_2]\!](w)$. Then $\langle C_1 \| C_2, w\rangle \longrightarrow^* w'$. By Lemma 2, there is a sequence $a_1, \ldots, a_{n'}$ of actions such that $w' = ([\![a_{n'}]\!] \circ \cdots \circ [\![a_1]\!])(w)$. By Lemma 5, the sequence $a_1, \ldots, a_{n'}$ can be decomposed into two subsequences $a_{L^1_1}, \ldots, a_{L^1_{n'_1}}$ and $a_{L^2_1}, \ldots, a_{L^2_{n'_2}}$ such that:

$$\langle C_1, w\rangle \longrightarrow^* ([\![a_{L^1_{n'_1}}]\!] \circ \cdots \circ [\![a_{L^1_1}]\!])(w), \quad \langle C_2, w\rangle \longrightarrow^* ([\![a_{L^2_{n'_2}}]\!] \circ \cdots \circ [\![a_{L^2_1}]\!])(w).$$

Now we define:

$$w^* = ([\![a_{L^2_{n'_2}}]\!] \circ \cdots \circ [\![a_{L^2_1}]\!] \circ [\![a_{L^1_{n'_1}}]\!] \circ \cdots \circ [\![a_{L^1_1}]\!])(w).$$

Then $w^* \in [\![C_1 \| C_2]\!](w)$. By Lemma 4, we obtain $\langle C_1 \| C_2, w\rangle \longrightarrow^* w^*$. We now show $m_{w'}(v) = m_{w^*}(v)$ for all $v \in \mathsf{Var}$ as follows. If $v \in \mathsf{Var}(C_1)$ then no substitution in C_2 updates the value of v, since $\mathsf{upd}(C_2) \cap \mathsf{Var}(C_1) = \emptyset$. Hence, $m_{w'}(v) = m_{([\![a_{L^1_{n'_1}}]\!] \circ \cdots \circ [\![a_{L^1_1}]\!])(w))}(v) = m_{w^*}(v)$. Symmetrically, if $v \in \mathsf{Var}(C_2)$ then $m_{w'}(v) = m_{([\![a_{L^2_{n'_2}}]\!] \circ \cdots \circ [\![a_{L^2_1}]\!])(w)}(v) = m_{w^*}(v)$. If v is a history variable, then the value of v does not depend on the order of actions, because every update increases v by 1. Thus, we conclude $m_{w'}(v) = m_{w^*}(v)$. For the other case, the program $C_1 \| C_2$ does not change the value of v, hence $m_{w'}(v) = m_{w^*}(v) = m_w(v)$.

Finally, since the test histories are multisets, we have $H_{w'} = H_{w^*}$. □

Lemma 7. *For any possible world $w \in \mathcal{W}$, any formula $\varphi \in \mathsf{Fml}$, and any interpretation function $\mathcal{I} : \mathsf{IntVar} \to \mathbb{Z}^*$, we have:*

$$[\![C_1; C_2]\!](w) \vDash^{\mathcal{I}} \varphi \ \text{ iff } \ [\![C_1 \| C_2]\!](w) \vDash^{\mathcal{I}} \varphi.$$

Proof. By Lemma 6, for any $w' \in [\![C_1 \| C_2]\!](w)$, there is a function $q_w : \mathcal{W} \to \mathcal{W}$ such that $q_w(w') \in [\![C_1; C_2]\!](w)$, $m_{q_w(w')} = m_{w'}$, and $H_{q_w(w')} = H_{w'}$.

By Lemma 3, it is sufficient to show the following statement:

$$\text{for all } w' \in [\![C_1 \| C_2]\!](w), \quad w' \vDash^{\mathcal{I}} \varphi \ \text{ iff } \ q_w(w') \vDash^{\mathcal{I}} \varphi.$$

We prove this by induction on the construction of the world w and the formula φ.

- Case $\varphi \equiv \eta(u_1, \ldots, u_k)$. Since $m_{w'} = m_{q_w(w')}$ and $V_{w'}(\eta) = V_{q_w(w')}(\eta)$,

$$w' \vDash^{\mathcal{I}} \varphi \ \text{ iff } \ ([\![u_1]\!]_{w'}, \ldots, [\![u_k]\!]_{w'}) \in V_{w'}(\eta)$$
$$\text{iff } \ ([\![u_1]\!]_{q_w(w')}, \ldots, [\![u_k]\!]_{q_w(w')}) \in V_{q_w(w')}(\eta)$$

iff $q_w(w') \models^{\mathcal{I}} \varphi$.

- Case $\varphi \equiv \mathbf{K}\varphi_1$. We prove the direction from left to right, that is,

  $$w' \models^{\mathcal{I}} \mathbf{K}\varphi_1 \text{ implies } q_w(w') \models^{\mathcal{I}} \mathbf{K}\varphi_1.$$

  We first recall the interpretation of $\mathbf{K}$.

  $$w' \models^{\mathcal{I}} \mathbf{K}\varphi_1 \text{ iff for all } w'_1 \in \mathcal{W},\ (w', w'_1) \in \mathcal{R} \text{ implies } w'_1 \models^{\mathcal{I}} \varphi_1.$$

  Assume that $w' \models^{\mathcal{I}} \mathbf{K}\varphi_1$. Let w'_1 be a world such that $(w', w'_1) \in \mathcal{R}$. Then $w'_1 \models^{\mathcal{I}} \varphi_1$. Let w_1 be a world such that $(w, w_1) \in \mathcal{R}$. Since $w' \in [\![C_1 \| C_2]\!](w)$, by Lemma 2, there is a sequence $a_1, \dots, a_{n'}$ of actions such that $w' = ([\![a_{n'}]\!] \circ \cdots \circ [\![a_1]\!])(w)$. Let $a_{L_1^1}, \dots, a_{L^1_{n'_1}}$ and $a_{L_1^2}, \dots, a_{L^2_{n'_2}}$ be the sequences of actions in C_1 and in C_2, respectively. By constructions in Lemma 6 (using Lemma 5), we then obtain:

  $$q_w(w') = ([\![a_{L^2_{n'_2}}]\!] \circ \cdots \circ [\![a_{L_1^2}]\!] \circ [\![a_{L^1_{n'_1}}]\!] \circ \cdots \circ [\![a_{L_1^1}]\!])(w),$$

  where the sequence $a_1, \dots, a_{n'}$ is decomposed into the subsequences $a_{L_1^1}, \dots, a_{L^1_{n'_1}}$ and $a_{L_1^2}, \dots, a_{L^2_{n'_2}}$.
  Since $C_1 \| C_2$ reads only observable variables, and $(w', w'_1) \in \mathcal{R}$ and $(w, w_1) \in \mathcal{R}$, we obtain $w'_1 = ([\![a_{n'}]\!] \circ \cdots \circ [\![a_1]\!])(w_1)$ and $w'_1 \in [\![C_1 \| C_2]\!](w_1)$. By induction hypothesis, we have $q_{w_1}(w'_1) \models^{\mathcal{I}} \varphi_1$.
  Let w'_2 be a world such that $(q_w(w'), w'_2) \in \mathcal{R}$. Then we have $w'_2 = ([\![a_{L^2_{n'_2}}]\!] \circ \cdots \circ [\![a_{L_1^2}]\!] \circ [\![a_{L^1_{n'_1}}]\!] \circ \cdots \circ [\![a_{L_1^1}]\!])(w_1)$. Since $w'_1 = ([\![a_{n'}]\!] \circ \cdots \circ [\![a_1]\!])(w_1)$, we obtain $q_{w_1}(w'_1) = w'_2$. Hence, we obtain $w'_2 \models^{\mathcal{I}} \varphi_1$ from $q_{w_1}(w'_1) \models^{\mathcal{I}} \varphi_1$.
  Since w'_2 is an arbitrary possible world such that $(q_w(w'), w'_2) \in \mathcal{R}$, we conclude $q_w(w') \models^{\mathcal{I}} \mathbf{K}\varphi_1$.
  The direction from right to left can be proved straightforwardly by Lemma 3.
- Cases $\varphi \equiv \neg\varphi_1$, $\varphi \equiv \varphi_1 \wedge \varphi_2$ and $\varphi \equiv \forall i.\varphi_1$. The statement is proved immediately by induction hypothesis. □

B.4. Proof for BHL's soundness

To prove BHL's soundness and relative completeness, we show the following lemma.

Lemma 8. *Let $\psi \in \mathsf{Fml}$, and $\mathcal{I}$ be any interpretation function over* IntVar. *Then:*

$$[\![\mathtt{skip}]\!](w) \models^{\mathcal{I}} \psi \quad \textit{iff} \quad w \models^{\mathcal{I}} \psi \tag{B.5}$$

$$[\![v := e]\!](w) \models^{\mathcal{I}} \psi \quad \textit{iff} \quad w \models^{\mathcal{I}} \psi[^{e}/_{v}] \tag{B.6}$$

$$[\![v := f_A(y)]\!](w) \models^{\mathcal{I}} \psi \quad \textit{iff} \quad w \models^{\mathcal{I}} \psi[^{f_A(y)}/_{v}, {}^{h_{y,A}+1}/_{h_{y,A}}]. \tag{B.7}$$

Proof of (B.5) in Lemma 8. Let $w' \overset{\text{def}}{=} [\![\mathtt{skip}]\!](w) = w; (m_w, \mathtt{skip}, H_w)$. We prove the statement by induction on ψ as follows.

- Case $\psi \equiv \eta(u_1, \dots, u_k)$. Since $V_w(\eta) = V_{w'}(\eta)$ and $m_w = m_{w'}$, we obtain:

  $$\begin{aligned}
  [\![\mathtt{skip}]\!](w) \models^{\mathcal{I}} \eta(u_1, \dots, u_k) \quad &\text{iff} \quad w' \models^{\mathcal{I}} \eta(u_1, \dots, u_k) \\
  &\text{iff} \quad ([\![u_1]\!]^{\mathcal{I}}_{w'}, \dots, [\![u_k]\!]^{\mathcal{I}}_{w'}) \in V_{w'}(\eta) \\
  &\text{iff} \quad ([\![u_1]\!]^{\mathcal{I}}_{w}, \dots, [\![u_k]\!]^{\mathcal{I}}_{w}) \in V_{w}(\eta) \\
  &\text{iff} \quad w \models^{\mathcal{I}} \eta(u_1, \dots, u_k).
  \end{aligned}$$

- Case $\psi \equiv \mathbf{K}\varphi$. We have:

  $$\begin{aligned}
  [\![\mathtt{skip}]\!](w) \models^{\mathcal{I}} \mathbf{K}\varphi \quad &\text{iff} \quad w' \models^{\mathcal{I}} \mathbf{K}\varphi \\
  &\text{iff for all } w_1 \in \mathcal{W},\ (w', w_1) \in \mathcal{R} \text{ implies } w_1 \models^{\mathcal{I}} \varphi \\
  &\text{iff for all } w_2 \in \mathcal{W},\ (w, w_2) \in \mathcal{R} \text{ implies } w_2 \models^{\mathcal{I}} \varphi. \qquad (\dagger)
  \end{aligned}$$

  The last equivalence (†) is proved as follows. To show the direction from left to right, let $(w, w_2) \in \mathcal{R}$. We take $w_1 = [\![\mathtt{skip}]\!](w_2) = w_2; (m_{w_2}, \mathtt{skip}, H_{w_2})$. We then obtain $(w', w_1) \in \mathcal{R}$, hence $w_1 \models^{\mathcal{I}} \varphi$. By induction hypothesis, we conclude $w_2 \models^{\mathcal{I}} \varphi$. To show the other direction, let $(w', w_1) \in \mathcal{R}$. We take $w_2 = w_1[0]; \cdots; w_1[\mathsf{len}(w_1) - 2]$. We then obtain $(w, w_2) \in \mathcal{R}$ and $[\![\mathtt{skip}]\!](w_2) = w_1$. Hence $w_2 \models^{\mathcal{I}} \varphi$. By induction hypothesis, we conclude $w_1 \models^{\mathcal{I}} \varphi$.
- Cases $\psi \equiv \neg\varphi$, $\psi \equiv \varphi_1 \wedge \varphi_2$, $\psi \equiv \forall i.\varphi$. Immediate by induction hypothesis. □

Proof of (B.6) in Lemma 8. Let $w' \stackrel{\text{def}}{=} [\![v := e]\!](w) = w; (m_w[v \mapsto [\![e]\!]_w], v := e, H_w)$. We prove the statement by induction on ψ as follows.

- Case $\psi \equiv \eta(u_1, \dots, u_k)$. For any term u, we have $[\![u]\!]^{\mathcal{I}}_{w'} = [\![u_1[e/v]]\!]^{\mathcal{I}}_{w}$. Since $V_w(\eta) = V_{w'}(\eta)$ and $m_{w'} = m_w[v \mapsto [\![e]\!]_w]$, we have:

$$\begin{aligned}
[\![v := e]\!](w) \vDash^{\mathcal{I}} \eta(u_1, \dots, u_k) \quad &\text{iff} \quad w' \vDash^{\mathcal{I}} \eta(u_1, \dots, u_k) \\
&\text{iff} \quad ([\![u_1]\!]^{\mathcal{I}}_{w'}, \dots, [\![u_k]\!]^{\mathcal{I}}_{w'}) \in V_{w'}(\eta) \\
&\text{iff} \quad ([\![u_1[e/v]]\!]^{\mathcal{I}}_{w}, \dots, [\![u_k[w/v]]\!]^{\mathcal{I}}_{w}) \in V_{w}(\eta) \\
&\text{iff} \quad w \vDash^{\mathcal{I}} \eta(u_1, \dots, u_k)[e/v].
\end{aligned}$$

- Case $\psi \equiv \mathbf{K}\varphi$. We have:

$$\begin{aligned}
[\![v := e]\!](w) \vDash^{\mathcal{I}} \mathbf{K}\varphi \ \text{iff} \ \ w' \vDash^{\mathcal{I}} \mathbf{K}\varphi \\
&\text{iff} \ \ \text{for all } w_1 \in \mathcal{W},\ (w', w_1) \in \mathcal{R} \text{ implies } w_1 \vDash^{\mathcal{I}} \varphi \\
&\text{iff} \ \ \text{for all } w_2 \in \mathcal{W},\ (w, w_2) \in \mathcal{R} \text{ implies } w_2 \vDash^{\mathcal{I}} \varphi[e/v]. \quad (\dagger)
\end{aligned}$$

  The last equivalence (†) is derived as with the proof of (B.5). To show the direction from left to right, let $(w, w_2) \in \mathcal{R}$ and $w_1 = [\![v := e]\!](w_2) = w_2; (m_{w_2}[v \mapsto [\![e]\!]_{w_2}], v := e, H_{w_2})$. Then $(w', w_1) \in \mathcal{R}$. By induction hypothesis, we conclude $w_2 \vDash^{\mathcal{I}} \varphi[e/v]$. To show the other direction, let $(w', w_1) \in \mathcal{R}$ and $w_2 = w_1[0]; \cdots; w_1[\mathsf{len}(w_1) - 2]$. Then $[\![v := e]\!](w_2) = w_1$ and $(w, w_2) \in \mathcal{R}$. By induction hypothesis, we conclude $w_1 \vDash^{\mathcal{I}} \varphi$.
- Cases $\psi \equiv \neg\varphi$, $\psi \equiv \varphi_1 \wedge \varphi_2$, $\psi \equiv \forall i.\varphi$. Immediate by induction hypothesis. □

Proof of (B.7) in Lemma 8. We prove (B.7) by induction on ψ as follows. We define:

$$w' \stackrel{\text{def}}{=} [\![v := f_A(y)]\!](w) = w; (m_w[v \mapsto [\![f_A(y)]\!]_w, h_{y,A} \mapsto [\![h_{y,A}]\!]_w + 1], a', H')$$

where $a' = v := f_A(y)$ and $H' = H_w \uplus \{m_w(y) \mapsto \{A\}\}$.

- Case $\psi \equiv \eta(u_1, \dots, u_k)$. As with the proof of (B.6), we obtain:

$$[\![v := f_A(y)]\!](w) \vDash^{\mathcal{I}} \eta(u_1, \dots, u_k) \quad \text{iff} \quad w \vDash^{\mathcal{I}} \eta(u_1, \dots, u_k)[{}^{f_A(y)}/_{v}, {}^{h_{y,A}+1}/_{h_{y,A}}].$$

- Case $\psi \equiv \mathbf{K}\varphi$. We have:

$$\begin{aligned}
[\![v := f_A(y)]\!](w) \vDash^{\mathcal{I}} \mathbf{K}\varphi \ \ \text{iff} \ \ w' \vDash^{\mathcal{I}} \mathbf{K}\varphi \\
&\text{iff} \ \ \text{for all } w_1 \in \mathcal{W},\ \ (w', w_1) \in \mathcal{R} \text{ implies } w_1 \vDash^{\mathcal{I}} \varphi \\
&\text{iff} \ \ \text{for all } w_2 \in \mathcal{W},\ \ (w, w_2) \in \mathcal{R} \text{ implies } w_2 \vDash^{\mathcal{I}} \varphi[{}^{f_A(y)}/_{v}, {}^{h_{y,A}+1}/_{h_{y,A}}]. \quad (\dagger)
\end{aligned}$$

  The last equivalence (†) is proved in a similar way to (B.6). To show the direction from left to right, let $(w, w_2) \in \mathcal{R}$. We define w_1 by:

$$w_1 \stackrel{\text{def}}{=} [\![v := f_A(y)]\!](w_2) = w_2; \left(\begin{smallmatrix} m_{w_2}[v \mapsto [\![f_A(y)]\!]_{w_2}, h_{y,A} \mapsto [\![h_{y,A}]\!]_{w_2} + 1], \\ v := f_A(y),\ H_{w_2} \uplus \{m_{w_2}(y) \mapsto \{A\}\} \end{smallmatrix} \right).$$

  Then $(w', w_1) \in \mathcal{R}$. By induction hypothesis, we conclude $w_2 \vDash^{\mathcal{I}} \varphi[{}^{f_A(y)}/_{v}, {}^{h_{y,A}+1}/_{h_{y,A}}]$. To show the other direction, let $(w', w_1) \in \mathcal{R}$ and $w_2 = w_1[0]; \cdots; w_1[\mathsf{len}(w_1) - 2]$. Then $[\![v := f_A(y)]\!](w_2) = w_1$ and $(w, w_2) \in \mathcal{R}$. By induction hypothesis, we conclude $w_1 \vDash^{\mathcal{I}} \varphi$.
- Cases $\psi \equiv \neg\varphi$, $\psi \equiv \varphi_1 \wedge \varphi_2$, $\psi \equiv \forall i.\varphi$. Immediate by induction hypothesis. □

Now we prove the soundness of BHL as follows.

Theorem 1 *(Soundness). Every derivable judgment is valid.*

Proof for Theorem 1. We obtain the validity of the axioms and rules for basic constructs in Fig. 2 as usual.

We show the validity of HIST as follows. Recall that the precondition in HIST is $\psi_{\mathsf{pre}} \stackrel{\text{def}}{=} \psi[v \mapsto f_A(y), h_{y,A} \mapsto (h_{y,A} + 1)]$. Let $w = (m_w, a_w, H_w)$ be a possible world such that $w \vDash \psi_{\mathsf{pre}}$. Let $w' \stackrel{\text{def}}{=} [\![v := f_A(y)]\!](w)$. By the semantics, we have $m_{w'} = m_w[v \mapsto [\![f_A(y)]\!]_{m_w}, h_{y,A} \mapsto [\![h_{y,A} + 1]\!]_{m_w}]$. By Lemma 8, we obtain $w' \vDash \psi$. Therefore, $(\Gamma^{\mathsf{inv}}, \Gamma^{\mathsf{obs}}) \vdash \{\psi_{\mathsf{pre}}\}\ v := f_A(y)\ \{\psi\}$ is valid.

We show the validity of PAR as follows. Assume that $(\Gamma^{\mathsf{inv}}, \Gamma^{\mathsf{obs}}) \vdash \{\psi\}\ C_1; C_2\ \{\psi'\}$. Let w be a world such that $w \vDash \psi$. Then $[\![C_1; C_2]\!](w) \vDash \psi'$. By Lemma 7 in Appendix B.3, we obtain $[\![C_1 \parallel C_2]\!](w) \vDash \psi'$. Hence, $(\Gamma^{\mathsf{inv}}, \Gamma^{\mathsf{obs}}) \vdash \{\psi\}\ C_1 \parallel C_2\ \{\psi'\}$.

Therefore, we obtain BHL's soundness. □

B.5. Proof for BHL's relative completeness

To prove BHL's relative completeness (Theorem 2), we show the notions of extension and weakest preconditions as follows.

Definition 8 *(Extension).* For a Kripke model $\mathfrak{M}$ with a domain $\mathcal{W}$ and an interpretation function $\mathcal{I}$, we define the *extension* of a formula $\varphi \in \mathsf{Fml}$ by:

$$\varphi^{\mathcal{I}} \stackrel{\text{def}}{=} \{w \in \mathcal{W} \mid w \vDash^{\mathcal{I}} \varphi\}.$$

For a $\varphi \in \mathsf{Fml}$ and a set S of worlds, we write $S \vDash \varphi$ iff $w \vDash \varphi$ for all $w \in S$.

Definition 9 *(Weakest precondition).* The *weakest precondition* of a formula φ w.r.t. a program C in $\mathcal{I}$ is defined by:

$$\mathsf{wp}^{\mathcal{I}}(C, \varphi) \stackrel{\text{def}}{=} \{w \in \mathcal{W} \mid [\![C]\!](w) \vDash^{\mathcal{I}} \varphi\}.$$

To derive BHL's relative completeness, we show the expressiveness of ELHT.

Proposition 4 *(Expressiveness). The assertion language ELHT is expressive; i.e., for every program C and every formula $\varphi \in \mathsf{Fml}$, there is a formula $F \in \mathsf{Fml}$ such that $F^{\mathcal{I}} = \mathsf{wp}^{\mathcal{I}}(C, \varphi)$.*

Proof. Let $w \in \mathcal{W}$ and $\varphi \in \mathsf{Fml}$. By the definition of the weakest precondition, it is sufficient to prove that there is a formula $F_C^{\varphi} \in \mathsf{Fml}$ such that:

$$w \vDash^{\mathcal{I}} F_C^{\varphi} \text{ iff } [\![C]\!](w) \vDash^{\mathcal{I}} \varphi. \tag{B.8}$$

We show this by induction on the program C. The proof is analogous to [5] except for the case of parallel composition.

- Case $C \equiv \mathtt{skip}$. Let $F_C^{\varphi} \stackrel{\text{def}}{=} \varphi$. Then:

$$\begin{aligned} w \vDash^{\mathcal{I}} F_C^{\varphi} &\text{ iff } w \vDash^{\mathcal{I}} \varphi && \text{(by Lemma 8 (B.5))} \\ &\text{iff } [\![\mathtt{skip}]\!](w) \vDash^{\mathcal{I}} \varphi. \end{aligned}$$

- Case $C \equiv v := e$. Let $F_C^{\varphi} \stackrel{\text{def}}{=} \varphi[^{e}/_{v}]$. Then:

$$\begin{aligned} w \vDash^{\mathcal{I}} F_C^{\varphi} &\text{ iff } w \vDash^{\mathcal{I}} \varphi[^{e}/_{v}] && \text{(by Lemma 8 (B.6))} \\ &\text{iff } [\![v := e]\!](w) \vDash^{\mathcal{I}} \varphi. \end{aligned}$$

- Case $C \equiv v := f_A(y)$. Let $F_C^{\varphi} \stackrel{\text{def}}{=} \varphi[^{f_A(y)}/_{v}, ^{(h_{y,A}+1)}/_{h_{y,A}}]$. Then we have:

$$\begin{aligned} w \vDash^{\mathcal{I}} F_C^{\varphi} &\text{ iff } w \vDash^{\mathcal{I}} \varphi[^{f_A(y)}/_{v}, ^{(h_{y,A}+1)}/_{h_{y,A}}] && \text{(by Lemma 8 (B.7))} \\ &\text{iff } [\![v := f_A(y)]\!](w) \vDash^{\mathcal{I}} \varphi. \end{aligned}$$

- Case $C \equiv C_1; C_2$. Let $F_C^{\varphi} \stackrel{\text{def}}{=} F_{C_1}^{F_{C_2}^{\varphi}}$. Then:

$$\begin{aligned} w \vDash^{\mathcal{I}} F_C^{\varphi} &\text{ iff } w \vDash^{\mathcal{I}} F_{C_1}^{F_{C_2}^{\varphi}} \\ &\text{iff } [\![C_1]\!](w) \vDash^{\mathcal{I}} F_{C_2}^{\varphi} && \text{(by induction hypothesis)} \\ &\text{iff } [\![C_2]\!]([\![C_1]\!](w)) \vDash^{\mathcal{I}} \varphi && \text{(by induction hypothesis)} \\ &\text{iff } [\![C_1; C_2]\!](w) \vDash^{\mathcal{I}} \varphi. \end{aligned}$$

- Case $C \equiv C_1 \| C_2$. Let $F_C^{\varphi} \stackrel{\text{def}}{=} F_{C_1}^{F_{C_2}^{\varphi}}$. Then:

$$\begin{aligned} w \vDash^{\mathcal{I}} F_C^{\varphi} &\text{ iff } w \vDash^{\mathcal{I}} F_{C_1}^{F_{C_2}^{\varphi}} && \text{(by applying the case } C = C_1; C_2) \\ &\text{iff } [\![C_1; C_2]\!](w) \vDash^{\mathcal{I}} \varphi && \text{(by Lemma 7)} \\ &\text{iff } [\![C_1 \| C_2]\!](w) \vDash^{\mathcal{I}} \varphi. \end{aligned}$$

- Case $C \equiv \texttt{if } e \texttt{ then } C_1 \texttt{ else } C_2$. Let $F^{\varphi}_{C} \stackrel{\text{def}}{=} (e \wedge F^{\varphi}_{C_1}) \vee (\neg e \wedge F^{\varphi}_{C_2})$. Then:

$$\begin{aligned} w \vDash^{\mathcal{I}} F^{\varphi}_{C} \text{ iff } & w \vDash^{\mathcal{I}} (e \wedge F^{\varphi}_{C_1}) \vee (\neg e \wedge F^{\varphi}_{C_2}) \\ \text{iff } & \text{either } \left([\![e]\!]_w = \texttt{true} \text{ and } w \vDash^{\mathcal{I}} F^{\varphi}_{C_1} \right) \text{ or } \left([\![e]\!]_w = \texttt{false} \text{ and } w \vDash^{\mathcal{I}} F^{\varphi}_{C_2} \right) \\ \text{iff } & \text{either } \left([\![e]\!]_w = \texttt{true} \text{ and } [\![C_1]\!](w) \vDash^{\mathcal{I}} \varphi \right) \text{ or } \left([\![e]\!]_w = \texttt{false} \text{ and } [\![C_2]\!](w) \vDash^{\mathcal{I}} \varphi \right) \quad \text{(by induction hypotheses)} \\ \text{iff } & [\![C]\!](w) \vDash^{\mathcal{I}} \varphi. \end{aligned}$$

- Case $C \equiv \texttt{loop } e \texttt{ do } C'$. The way of the proof is similar to [5]. By the semantics of Prog, $[\![C]\!](w) \vDash^{\mathcal{I}} \varphi$ is logically equivalent to:

$$\begin{aligned} & \forall k\ \forall w_0, \dots, w_k \in \mathcal{W}.\quad w = w_0 \text{ and} \\ & \qquad \forall i = 0, \dots, k. \left(w_i \vDash^{\mathcal{I}} e \text{ and } [\![C']\!](w_i) = w_{i+1} \right) \text{ implies } w_k \vDash^{\mathcal{I}} e \vee \varphi. \end{aligned} \tag{B.9}$$

To describe this using the assertion language ELHT, we replace the possible worlds w_i $(i = 0, \dots, k)$ with equivalent assertions as follows. Let $\overline{v} = (v_1, \dots, v_l)$ be all observable and invisible variables occurring in C or φ. Then $\overline{v} \cap \mathsf{IntVar} = \emptyset$. For $i = 0, \dots, k$ and $j = 1, \dots, l$, let $s_{ij} = w_i(v_j)$ and $\overline{s_i} = (s_{i1}, \dots, s_{il}) \in \mathbb{Z}^l$. We write $\varphi[\overline{s_i}/\overline{v}]$ for the assertion obtained by the simultaneous substitution of $\overline{s_i}$ for $\overline{v}$ in φ. Then each w_i can be converted into the equivalent substitution $[\overline{s_i}/\overline{v}]$ as follows: for any interpretation function $\mathcal{I}$,

$$w_i \vDash^{\mathcal{I}} \varphi \quad \text{iff} \quad \vDash^{\mathcal{I}} \varphi[\overline{s_i}/\overline{v}] \quad \text{iff} \quad w \vDash^{\mathcal{I}} \varphi[\overline{s_i}/\overline{v}]. \tag{B.10}$$

Then neither observable nor invisible variable occurs in $\varphi[\overline{s_i}/\overline{v}]$.
By (B.10), we can replace the worlds $w_0, \dots, w_k$ in (B.9) with their corresponding substitutions $[\overline{s_0}/\overline{v}], \dots, [\overline{s_k}/\overline{v}]$. Thus, (B.9) is logically equivalent to:

$$\begin{aligned} & \forall k\ \forall \overline{s_0}, \dots, \overline{s_k} \in \mathbb{Z}^l.\quad w \vDash^{\mathcal{I}} (\overline{v} = \overline{s_0}) \text{ and} \\ & \qquad \forall i = 0, \dots, k. \left(w \vDash^{\mathcal{I}} e[\overline{s_i}/\overline{v}] \text{ and } [\![C']\!](w_i) = w_{i+1} \right) \text{ implies } w \vDash^{\mathcal{I}} (e \vee \varphi)[\overline{s_k}/\overline{v}]. \end{aligned} \tag{B.11}$$

To express $[\![C']\!](w_i) = w_{i+1}$ as a formula, we derive:

$$\begin{aligned} [\![C']\!](w_i) = w_{i+1} \text{ iff } & [\![C']\!](w_i) \neq \emptyset \text{ and } [\![C']\!](w_i) \vDash^{\mathcal{I}} \overline{v} = \overline{s_{i+1}} && \text{(by induction hypothesis)} \\ \text{iff } & w_i \vDash^{\mathcal{I}} \neg F^{\texttt{false}}_{C'} \text{ and } w_i \vDash^{\mathcal{I}} F^{(\overline{v} = \overline{s_{i+1}})}_{C'} && \text{(by (B.10))} \\ \text{iff } & w \vDash^{\mathcal{I}} \neg F^{\texttt{false}}_{C'}[\overline{s_i}/\overline{v}] \wedge F^{(\overline{v} = \overline{s_{i+1}})}_{C'}[\overline{s_i}/\overline{v}]. \end{aligned} \tag{B.12}$$

Now we define F^{φ}_{C} by:

$$\begin{aligned} F^{\varphi}_{C} \stackrel{\text{def}}{=}\ & \forall k\ \forall \overline{s_0}, \dots, \overline{s_k} \in \mathbb{Z}^l. \\ & \left((\overline{v} = \overline{s_0}) \wedge\ \forall i = 0, \dots, k. (e \wedge \neg F^{\texttt{false}}_{C'} \wedge F^{(\overline{v} = \overline{s_{i+1}})}_{C'})[\overline{s_i}/\overline{v}] \right) \\ & \to (e \vee \varphi)[\overline{s_k}/\overline{v}]. \end{aligned} \tag{B.13}$$

By (B.12) and (B.13), $w \vDash^{\mathcal{I}} F^{\varphi}_{C}$ is logically equivalent to (B.11). Hence,

$$\begin{aligned} [\![C]\!](w) \vDash^{\mathcal{I}} \varphi \quad & \text{iff} \quad \text{(B.9)} \\ & \text{iff} \quad \text{(B.11)} \\ & \text{iff} \quad w \vDash^{\mathcal{I}} F^{\varphi}_{C}. \end{aligned}$$

Therefore, we obtain (B.8). □

Lemma 9. *For a program C and a formula $\varphi \in \mathsf{Fml}$, let F^{φ}_{C} be a formula expressing the weakest precondition, i.e., $(F^{\varphi}_{C})^{\mathcal{I}} = \mathsf{wp}^{\mathcal{I}}(C, \varphi)$. Then we obtain $\Gamma \vdash \{F^{\varphi}_{C}\}\ C\ \{\varphi\}$.*

Proof. We show the lemma by induction on the program C as follows.

- Case $C \equiv \texttt{skip}$. By applying the rule (Conseq) to $\vDash F^{\varphi}_{C} \to \varphi$ and the axiom (Skip) $\Gamma \vdash \{\varphi\}\ C\ \{\varphi\}$, we obtain $\Gamma \vdash \{F^{\varphi}_{C}\}\ C\ \{\varphi\}$.
- Case $C \equiv v := e$. By applying the rule (Conseq) to $\vDash F^{\varphi}_{C} \to \varphi[v \mapsto e]$ and the axiom (UpdVar) $\Gamma \vdash \{\varphi[v \mapsto e]\}\ C\ \{\varphi\}$, we obtain $\Gamma \vdash \{F^{\varphi}_{C}\}\ C\ \{\varphi\}$.
- Case $C \equiv v := f_A(y)$. By applying the rule (Conseq) to $\vDash F^{\varphi}_{C} \to \varphi[v \mapsto f_A(y), h_{y,A} \mapsto h_{y,A} + 1]$ and the axiom (Hist), we obtain $\Gamma \vdash \{F^{\varphi}_{C}\}\ C\ \{\varphi\}$.

- Case $C \equiv C_1; C_2$. By induction hypothesis, we have:

$$\Gamma \vdash \{F^{F^{\varphi}_{C_2}}_{C_1}\}\ C_1\ \{F^{\varphi}_{C_2}\} \quad \text{and} \quad \Gamma \vdash \{F^{\varphi}_{C_2}\}\ C_2\ \{\varphi\}.$$

  By applying (Seq), we obtain $\Gamma \vdash \{F^{F^{\varphi}_{C_2}}_{C_1}\}\ C_1; C_2\ \{\varphi\}$. Hence by applying (Conseq) with $\vDash F^{\varphi}_{C_1;C_2} \to F^{F^{\varphi}_{C_2}}_{C_1}$, we conclude $\Gamma \vdash \{F^{\varphi}_{C_1;C_2}\}\ C_1; C_2\ \{\varphi\}$.
- Case $C \equiv C_1 \| C_2$. By applying the case of the sequential composition $C_1; C_2$, we have $\Gamma \vdash \{F^{\varphi}_{C_1;C_2}\}\ C_1; C_2\ \{\varphi\}$. By applying the rules (Par) and (Conseq) with $\vDash F^{\varphi}_{C_1\|C_2} \to F^{\varphi}_{C_1;C_2}$, we conclude $\Gamma \vdash \{F^{\varphi}_{C_1\|C_2}\}\ C_1 \| C_2\ \{\varphi\}$.
- Case $C \equiv \texttt{if}\ e\ \texttt{then}\ C_1\ \texttt{else}\ C_2$. By induction hypothesis, we have:

$$\Gamma \vdash \{F^{\varphi}_{C_1}\}\ C_1\ \{\varphi\} \quad \text{and} \quad \Gamma \vdash \{F^{\varphi}_{C_2}\}\ C_2\ \{\varphi\}.$$

  By applying (Conseq), we have $\Gamma \vdash \{e \wedge F^{\varphi}_{C_1}\}\ C_1\ \{\varphi\}$ and $\Gamma \vdash \{\neg e \wedge F^{\varphi}_{C_2}\}\ C_2\ \{\varphi\}$. Then, by applying the rules (If) and (Conseq) with:

$$\vDash F^{\varphi}_{\texttt{if}\ e\ \texttt{then}\ C_1\ \texttt{else}\ C_2} \to (e \wedge F^{\varphi}_{C_1}) \vee (\neg e \wedge F^{\varphi}_{C_2}),$$

  we conclude:

$$\Gamma \vdash \{F^{\varphi}_{\texttt{if}\ e\ \texttt{then}\ C_1\ \texttt{else}\ C_2}\}\ \texttt{if}\ e\ \texttt{then}\ C_1\ \texttt{else}\ C_2\ \{\varphi\}.$$

- Case $C \equiv \texttt{loop}\ e\ \texttt{do}\ C'$. Let $w \in \mathcal{W}$ and $\mathcal{I}$ be an interpretation function. Then:

$$\begin{aligned}
w \vDash^{\mathcal{I}} F^{\varphi}_{\texttt{loop}\ e\ \texttt{do}\ C'} \quad &\text{iff} \quad [\![\texttt{loop}\ e\ \texttt{do}\ C']\!](w) \vDash^{\mathcal{I}} \varphi \\
&\text{iff} \quad \text{either } ([\![e]\!](w) = \texttt{true} \ \text{ and } \ [\![C'; \texttt{loop}\ e\ \texttt{do}\ C']\!](w) \vDash^{\mathcal{I}} \varphi) \\
&\qquad \text{or} \quad ([\![e]\!](w) = \texttt{false} \ \text{ and } \ w \vDash^{\mathcal{I}} \varphi) \\
&\text{iff} \quad \text{either } ([\![e]\!](w) = \texttt{true} \ \text{ and } \ [\![C']\!](w) \vDash^{\mathcal{I}} F^{\varphi}_{\texttt{loop}\ e\ \texttt{do}\ C'}) \\
&\qquad \text{or} \quad ([\![e]\!](w) = \texttt{false} \ \text{ and } \ w \vDash^{\mathcal{I}} \varphi) \\
&\text{iff} \quad w \vDash^{\mathcal{I}} e \wedge F^{F^{\varphi}_{\texttt{loop}\ e\ \texttt{do}\ C'}}_{C'} \quad \text{or} \quad w \vDash^{\mathcal{I}} \neg e \wedge \varphi.
\end{aligned}$$

  This implies:

$$\vDash e \wedge F^{\varphi}_{\texttt{loop}\ e\ \texttt{do}\ C'} \to F^{F^{\varphi}_{\texttt{loop}\ e\ \texttt{do}\ C'}}_{C'} \tag{B.14}$$

$$\vDash \neg e \wedge F^{\varphi}_{\texttt{loop}\ e\ \texttt{do}\ C'} \to \varphi. \tag{B.15}$$

  By induction hypothesis, we have:

$$\Gamma \vdash \{F^{F^{\varphi}_{\texttt{loop}\ e\ \texttt{do}\ C'}}_{C'}\}\ C'\ \{F^{\varphi}_{\texttt{loop}\ e\ \texttt{do}\ C'}\}.$$

  By applying (Conseq) with (B.14), we have:

$$\Gamma \vdash \{e \wedge F^{\varphi}_{\texttt{loop}\ e\ \texttt{do}\ C'}\}\ C'\ \{F^{\varphi}_{\texttt{loop}\ e\ \texttt{do}\ C'}\}.$$

  By applying (Loop), we obtain:

$$\Gamma \vdash \{F^{\varphi}_{\texttt{loop}\ e\ \texttt{do}\ C'}\}\ \texttt{loop}\ e\ \texttt{do}\ C'\ \{\neg e \wedge F^{\varphi}_{\texttt{loop}\ e\ \texttt{do}\ C'}\}.$$

  By applying (Conseq) with (B.15), we conclude:

$$\Gamma \vdash \{F^{\varphi}_{\texttt{loop}\ e\ \texttt{do}\ C'}\}\ \texttt{loop}\ e\ \texttt{do}\ C'\ \{\varphi\}. \quad \square$$

Finally, we prove the relative completeness of BHL as follows.

Theorem 2 *(Relative completeness). Every valid judgment is derivable except for the proofs for assertions.*

Proof of Theorem 2. We assume the validity of a judgment $\Gamma \vDash \{\psi\}\ C\ \{\varphi\}$. Let w be a world such that $w \vDash \psi$. By the validity of the judgment, we have $w \in \mathsf{wp}^{\mathcal{I}}(C, \varphi)$.

By Proposition 4, there exists a formula F^{φ}_{C} that expresses the weakest precondition, that is, $(F^{\varphi}_{C})^{\mathcal{I}} = \mathsf{wp}^{\mathcal{I}}(C, \varphi)$ for any interpretation function $\mathcal{I}$. Thus, it follows from $w \vDash \psi$ and $w \in \mathsf{wp}^{\mathcal{I}}(C, \varphi)$ that $w \vDash F^{\varphi}_{C}$. Hence $\Gamma \vDash \psi \to F^{\varphi}_{C}$.

By Lemma 9, we obtain $\Gamma \vdash \{F_C^{\varphi}\}\ C\ \{\varphi\}$. By applying the rule (Conseq) to $\Gamma \vDash \psi \to F_C^{\varphi}$ and $\Gamma \vdash \{F_C^{\varphi}\}\ C\ \{\varphi\}$, we obtain $\Gamma \vdash \{\psi\}\ C\ \{\varphi\}$. □

References

[1] T.A. Lang, D.G. Altman, Statistical Analyses and Methods in the Published Literature: The SAMPL Guidelines, John Wiley & Sons, Ltd, 2014, pp. 264–274, Ch. 25.
[2] R.L. Wasserstein, N.A. Lazar, The ASA statement on p-values: context, process, and purpose, Am. Stat. 70 (2) (2016) 129–133, https://doi.org/10.1080/00031305.2016.1154108.
[3] Y. Kawamoto, T. Sato, K. Suenaga, Formalizing statistical beliefs in hypothesis testing using program logic, in: Proc. KR'21, 2021, pp. 411–421.
[4] C.A.R. Hoare, An axiomatic basis for computer programming, Commun. ACM 12 (10) (1969) 576–580, https://doi.org/10.1145/363235.363259.
[5] G. Winskel, The Formal Semantics of Programming Languages—An Introduction, The MIT Press, 1993.
[6] K.R. Apt, E. Olderog, Fifty years of hoare's logic, Form. Asp. Comput. 31 (6) (2019) 751–807, https://doi.org/10.1007/s00165-019-00501-3.
[7] J.C. Reynolds, Separation logic: a logic for shared mutable data structures, in: Proc. LICS'02, IEEE Computer Society, 2002, pp. 55–74.
[8] K. Suenaga, I. Hasuo, Programming with infinitesimals: a while-language for hybrid system modeling, in: Proc. ICALP'11, Part II, in: LNCS, vol. 6756, Springer, 2011, pp. 392–403.
[9] J. den Hartog, E.P. de Vink, Verifying probabilistic programs using a Hoare like logic, Int. J. Found. Comput. Sci. 13 (3) (2002) 315–340, https://doi.org/10.1142/S012905410200114X.
[10] E. Atkinson, M. Carbin, Programming and reasoning with partial observability, Proc. ACM Program. Lang. 4 (OOPSLA) (2020) 200:1–200:28, https://doi.org/10.1145/3428268.
[11] G.H. von Wright, An Essay in Modal Logic, North-Holland Pub. Co., Amsterdam, 1951.
[12] J. Hintikka, Knowledge and Belief: An Introduction to the Logic of the Two Notions, Cornell University Press, 1962.
[13] R. Fagin, J. Halpern, Y. Moses, M. Vardi, Reasoning About Knowledge, The MIT Press, 1995.
[14] M. Burrows, M. Abadi, R.M. Needham, A logic of authentication, ACM Trans. Comput. Syst. 8 (1) (1990) 18–36, https://doi.org/10.1145/77648.77649.
[15] P.F. Syverson, S.G. Stubblebine, Group principals and the formalization of anonymity, in: World Congress on Formal Methods, vol. 1, 1999, pp. 814–833.
[16] F.D. Garcia, I. Hasuo, W. Pieters, P. van Rossum, Provable anonymity, in: Proc. FMSE, 2005, pp. 63–72.
[17] J.Y. Halpern, Reasoning About Uncertainty, The MIT Press, 2003.
[18] F. Huber, C. Schmidt-Petri, Degrees of Belief, vol. 342, Springer Science & Business Media, 2008.
[19] F. Bacchus, J.Y. Halpern, H.J. Levesque, Reasoning about noisy sensors and effectors in the situation calculus, Artif. Intell. 111 (1–2) (1999) 171–208, https://doi.org/10.1016/S0004-3702(99)00031-4.
[20] Y. Kawamoto, Statistical epistemic logic, in: The Art of Modelling Computational Systems: A Journey from Logic and Concurrency to Security and Privacy, in: LNCS, vol. 11760, Springer, 2019, pp. 344–362.
[21] Y. Kawamoto, Towards logical specification of statistical machine learning, in: Proc. SEFM, 2019, pp. 293–311, https://arxiv.org/pdf/1907.10327.
[22] Y. Kawamoto, An epistemic approach to the formal specification of statistical machine learning, Softw. Syst. Model. 20 (2) (2020) 293–310, https://doi.org/10.1007/s10270-020-00825-2.
[23] H. Van Ditmarsch, W. van Der Hoek, B. Kooi, Dynamic Epistemic Logic, vol. 337, Springer Science & Business Media, 2007.
[24] L. Zadeh, Fuzzy sets, Inf. Control 8 (3) (1965) 338–353, https://doi.org/10.1016/S0019-9958(65)90241-X.
[25] H.T. Nguyen, C.L. Walker, E.A. Walker, A First Course in Fuzzy Logic, 4th edition, Chapman & Hall/CRC, 2018.
[26] C. Eberhart, A. Yamada, S. Klikovits, S. Katsumata, T. Kobayashi, I. Hasuo, F. Ishikawa, Architecture-guided test resource allocation via logic, in: Proc. TAP'21, in: LNCS, vol. 12740, Springer, 2021, pp. 22–38.
[27] R. Reiter, A logic for default reasoning, Artif. Intell. 13 (1–2) (1980) 81–132, https://doi.org/10.1016/0004-3702(80)90014-4.
[28] H.E.K. Jr., C. Teng, Evaluating defaults, in: Proc. the 9th International Workshop on Non-Monotonic Reasoning (NMR 2002), 2002, pp. 257–264.
[29] H.E.K. Jr., C. Teng, Nonmonotonic logic and statistical inference, Comput. Intell. 22 (1) (2006) 26–51, https://doi.org/10.1111/j.1467-8640.2006.00272.x.
[30] H.E.K. Jr., C. Teng, Statistical inference as default reasoning, Int. J. Pattern Recognit. Artif. Intell. 13 (2) (1999) 267–283, https://doi.org/10.1142/S021800149900015X.
[31] R. Fagin, J.Y. Halpern, Y. Moses, M.Y. Vardi, Knowledge-based programs, in: Proc. PODC'95, ACM, 1995, pp. 153–163.
[32] N. Laverny, J. Lang, From knowledge-based programs to graded belief-based programs, part I: on-line reasoning*, Synth. 147 (2) (2005) 277–321, https://doi.org/10.1007/s11229-005-1350-1.
[33] S. Sardiña, Y. Lespérance, Golog speaks the BDI language, in: Proc. ProMAS'09, in: LNCS, vol. 5919, Springer, 2009, pp. 82–99.
[34] H.J. Levesque, R. Reiter, Y. Lespérance, F. Lin, R.B. Scherl, GOLOG: a logic programming language for dynamic domains, J. Log. Program. 31 (1–3) (1997) 59–83, https://doi.org/10.1016/S0743-1066(96)00121-5.
[35] M. Bratman, Intention, Plans, and Practical Reason, 1987.
[36] V. Belle, H.J. Levesque, ALLEGRO: belief-based programming in stochastic dynamical domains, in: Proc. IJCAI 2015, AAAI Press, 2015, pp. 2762–2769.
[37] R.V. Hogg, J.W. McKean, A.T. Craig, Introduction to Mathematical Statistics, Prentice Hall, 2004.
[38] G.K. Kanji, 100 Statistical Tests, Sage, 2006.
[39] F. Bretz, T. Hothorn, P. Westfall, Multiple Comparisons Using R, Chapman and Hall/CRC, 2010.
[40] H.R. Nielson, F. Nielson, in: Semantics with Applications: An Appetizer, in: Undergraduate Topics in Computer Science, Springer-Verlag, 2007.
[41] S.A. Cook, Soundness and completeness of an axiom system for program verification, SIAM J. Comput. 7 (1) (1978) 70–90, https://doi.org/10.1137/0207005.
[42] A. Platzer, Logical Foundations of Cyber-Physical Systems, Springer, 2018.
[43] R. Hähnle, M. Huisman, Deductive software verification: from pen-and-paper proofs to industrial tools, in: Computing and Software Science - State of the Art and Perspectives, in: Lecture Notes in Computer Science, vol. 10000, Springer, 2019, pp. 345–373.
[44] S.A. Kripke, The undecidability of monadic modal quantification theory, Math. Log. Q. 8 (2) (1962) 113–116, https://doi.org/10.1002/malq.19620080204.
[45] G.E. Hughes, M.J. Cresswell, A New Introduction to Modal Logic, Psychology Press, 1996.
[46] E. Sober, Evidence and Evolution: The Logic Behind the Science, Cambridge University Press, 2008.
[47] Y. Kawamoto, T. Sato, K. Suenaga, Formalizing statistical causality via modal logic, in: Proc. JELIA 2023, in: Lecture Notes in Computer Science, vol. 14281, Springer, 2023, pp. 681–696.
[48] J. Neyman, E.S. Pearson, On the problem of the most efficient tests of statistical hypotheses, Philos. Trans. R. Soc. Lond., Ser. A, Contain. Pap. Math. Phys. Character 231 (1933) 289–337.