University of Oxford
Department of Computer Science

Fast and Correct Variational Inference for Probabilistic Programming

Differentiability, Reparameterisation and Smoothing

submitted by
Dominik Wagner
Oriel College

on 15th November 2023

Supervisor
Prof. C.-H. Luke Ong

A thesis submitted for the degree of
DPhil in Computer Science

Michaelmas Term 2023

Abstract

Probabilistic programming is an innovative programming paradigm for posing and automatically solving Bayesian inference problems. In this thesis, we study the foundations of *fast yet correct* inference for probabilistic programming.

Many of the most successful inference techniques (e.g. Hamiltonian Monte Carlo or Stochastic Variational Inference) harness gradients of the so-called density function, which therefore needs to be differentiable at least almost everywhere. We resolve a question posed by Hongseok Yang by demonstrating the following: densities of almost surely terminating programs are differentiable almost everywhere.

Having established this property necessary for the correctness of gradient-based inference algorithms, we investigate variational inference, which frames posterior inference as an optimisation problem, in more detail. The dominant approach for stochastic optimisation in practice is stochastic gradient descent. In particular, a variant using the so-called reparameterisation gradient estimator exhibits low variance, resulting in fast convergence in a traditional statistics setting. Unfortunately, although having measure 0, discontinuities can compromise the correctness of this approach.

Therefore, we propose a *smoothed* interpretation parameterised by an accuracy coefficient and present type systems establishing technical pre-conditions. Thus, we can prove stochastic gradient descent with the reparameterisation gradient estimator to be correct when applied to the smoothed problem. Besides, via a uniform convergence result, we can solve the original problem up to any error tolerance by choosing an accuracy coefficient suitably.

Furthermore, rather than fixing an accuracy coefficient in advance (limiting the quality of the final solution), we propose a novel variant of stochastic gradient descent, *Diagonalisation Stochastic Gradient Descent*, which progressively enhances the accuracy of the smoothed approximation *during* optimisation, and we prove convergence to stationary points of the *unsmoothed* (original) objective.

An empirical evaluation reveals benefits of our approaches over the state of the art: our approaches are simple, fast and attain orders of magnitude reduction in work-normalised variance. Besides, Diagonalisation Stochastic Gradient Descent is more stable than standard stochastic gradient descent for a fixed-accuracy smoothing.

Finally, we show unbiasedness of the reparameterisation gradient estimator for *continuous but non-differentiable* models, and we propose a method based on higher-order logic to establish continuity in the presence of conditionals. We provide a sound and complete reduction for verifying continuity of models to a satisfiability problem, and we propose novel efficient randomised decision procedures.

Acknowledgements

First of all, I would like to express my deep gratitude to my supervisor, Luke Ong, who introduced me to the intriguing topic of probabilistic programming and who has always been an invaluable guide to the various aspects of life as a researcher. I am also grateful to Andrzej Murawski for detailed comments on parts of this thesis. Besides, I thank colleagues and friends at Oxford and NTU Singapore for stimulating conversations and collaborations. I would like to give special thanks to Maria Craciun, Basim Khajwal, Xuan-Bach Le, Carol Mak, Rolf Morel, Hugo Paquet and Fabian Zaiser. Moreover, I am sincerely grateful for the community at Oriel College, where I enjoyed tutoring very talented undergraduates. Being member of the "Shadow Senior Common Room" has been an absolute pleasure and has contributed enormously to my intellectual maturation. Finally, I am deeply indebted to my family for their continuous support.

Contents

Chapter 1

Introduction

1.1 Motivation and Overview

Probabilistic programming [van de Meent et al., 2018, Barthe et al., 2020b] is a programming paradigm which has the vision to make statistical methods, in particular Bayesian inference, accessible to a wide audience. This is achieved by a separation of concerns: the domain experts wishing to gain statistical insights focus on modelling, whilst the inference is performed automatically. Thus, users of probabilistic programming **(i)** encode their domain knowledge in program form; **(ii)** *condition* certain program variables based on observed data; and **(iii)** make a query. The resulting code is then passed to an *inference engine* which performs the necessary computation to answer the query, usually following a generic approximate Bayesian inference algorithm. (In some recent systems [Bingham et al., 2019, Cusumano-Towner et al., 2019] users can improve efficiency by writing their own inference code.)

In essence, probabilistic programming languages extend more traditional programming languages with constructs for **(i)** sampling to define the prior $p(\boldsymbol{z})$, and **(ii)** conditioning (**score** or **observe**) for the likelihood $p(\boldsymbol{x} \mid \boldsymbol{z})$. (The query **(iii)** can usually be encoded as the return value of the program.)

Example 1.1. Suppose we live in a place with tropical climate, where the temperature is always around 30 degrees centigrade. Today it is quite windy and we conjecture it may be colder. Therefore, we consult our phone's moderately reliable weather app and our quite unreliable household thermometer, which display $30.3°C$ and $28.7°C$, respectively. How should we update our belief about the real temperature? We model the scenario as follows:

```
-- prior
let z = sample 𝒩(30,4)
-- likelihood
    observe 30.3 from 𝒩(z, 1)
    observe 28.7 from 𝒩(z, 2)
-- query
in  z
```

This defines the prior density $p(z) = \mathcal{N}(z \mid 30, 4)$ and likelihood

$$p(x_1 = 30.3, x_2 = 28.7 \mid z) = \mathcal{N}(30.3 \mid z, 1) \cdot \mathcal{N}(28.7 \mid z, 2)$$

The task of inference is to derive the posterior $p(\boldsymbol{z} \mid \boldsymbol{x})$, which is in principle governed by Bayes' law:

$$p(\boldsymbol{z} \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid \boldsymbol{z}) \cdot p(\boldsymbol{z})}{\int p(\boldsymbol{x} \mid \boldsymbol{z}') \cdot p(\boldsymbol{z}') \, \mathrm{d}\boldsymbol{z}'}$$

Unfortunately, this typically cannot be obtained in closed form because the denominator $\int p(\boldsymbol{x} \mid \boldsymbol{z}') \cdot p(\boldsymbol{z}') \, \mathrm{d}\boldsymbol{z}'$, which is referred to as *marginal likelihood* or *model evidence*, is intractable in general. Therefore, approximation methods have been devised, which broadly fall into two categories: Markov chain Monte Carlo (MCMC), which yields a sequence of samples asymptotically approaching the true posterior, and variational inference, which frames posterior inference as an optimisation problem.

It is crucial to have access to reasoning principles in this context. The combination of the new primitives for sampling and observation with the traditional constructs of programming languages leads to a variety of new computational phenomena, and a major concern is the *correctness of inference*: given a query, will the algorithm converge, in some appropriate sense, to a correct answer? Whilst the paradigm was originally conceived in the context of statistics and Bayesian machine learning, probabilistic programming has in recent years proven to be a very fruitful subject for the programming language community. In a *universal* PPL (i.e. one whose underlying language is Turing-complete), correctness can be very subtle, indeed: the inference engine must account for a wide class of programs, going beyond the more well-behaved models found in many of the current statistical applications. Researchers have made significant theoretical contributions such as underpinning languages with rigorous (categorical) semantics [Staton et al., 2016, Staton, 2017, Heunen et al., 2017b, Vákár et al., 2019, Ehrhard et al., 2014, Dahlqvist and Kozen, 2020] and investigating the correctness of inference algorithms [Hur et al., 2015, Borgström et al., 2016, Lee et al., 2020b]. The latter were mostly designed in the context of "traditional" statistics and features such as conditionals, which are ubiquitous in programming, pose a major challenge for correctness.

Thus, the design of inference algorithms, and the associated correctness proofs, are quite delicate. It is well-known, for instance, that in its original version the popular lightweight Metropolis-Hastings algorithm [Wingate et al., 2011] contained a bug affecting the result of inference [Hur et al., 2015, Kiselyov, 2016].

Fortunately, research in this area benefits from decades of work on the semantics of programs with random features, starting with pioneering work by [Kozen, 1979] and [Saheb-Djahromi, 1978]. Both operational and denotational models have recently been applied to the validation of inference algorithms: see e.g. [Hur et al., 2015, Borgström et al., 2016] for the former and [Ścibior et al., 2017, Castellan and Paquet, 2019] for the latter. There are other approaches, using type systems (e.g. [Lew et al., 2019, Gorinova et al., 2022, Lew et al., 2023]) or abstract interpretation (e.g. [Lee et al., 2020b, Lee et al., 2023]).

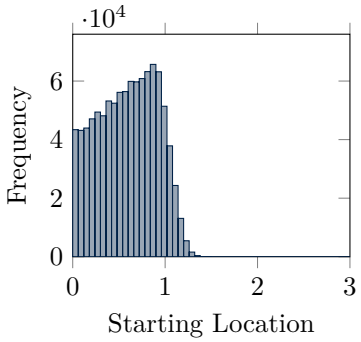
Probabilistic Programming: a (Running) Example

To get a better idea of the power of probabilistic programming to express complex models succinctly, we consider a variant of a random walk in $\mathbb{R}_{\geq 0}$ (see [Mak et al., 2021]).

The story is as follows: a pedestrian has gone on a walk on a certain semi-infinite street (i.e. extending infinitely on one side), where she may periodically change directions. Upon

```
-- returns total distance travelled
let rec walk start =
  if start <= 0 then
    0
  else
    -- each leg < 1km
    let step = sample uniform(0, 1) in
    if (sample bernoulli(0.5)) then
      -- go towards +infty
      step + walk (start+step)
    else
      -- go towards 0
      step + walk (start-step)
in
-- prior
let start    = sample uniform(0, 3) in
    distance = walk start in
-- likelihood
observe 1.1 from normal(distance, 0.1)
-- query
start
```

(a) Running example in pseudocode.



(b) Resulting histogram.

Figure 1.1: Inferring the starting point of a random walk on $\mathbb{R}_{\geq 0}$, in a PPL.

reaching the end of the street she has forgotten her starting point, only remembering that she started no more than 3km away. Thanks to an odometer, she knows the total distance she has walked is 1.1km, although there is a small margin of error. Her starting point can be inferred using probabilistic programming, via the program in Fig. 1.1a.

The function walk in Fig. 1.1a is a recursive simulation of the random walk: note that in this model a new direction is sampled after at most 1km. Once the pedestrian has travelled past 0 the function returns the total distance travelled. The rest of the program first specifies a *prior distribution* for the starting point, representing the pedestrian's belief — uniform distribution on $[0, 3]$ — before observing the distance measured by the odometer. After drawing a value for start the program simulates a random walk, and the execution is weighted (via score) according to how close distance is to the observed value of 1.1. The return value is our query: it indicates that we are interested in the *posterior* distribution on the starting point.

The histogram in Fig. 1.1b is obtained by sampling repeatedly from the posterior of a Python model of our running example. It shows the mode of the pedestrian's starting point to be around the 0.8km mark.

Program Traces

Inference algorithms in probabilistic programming are often based on the concept of *program trace*, because the operational behaviour of a program is parameterised by the

sequence of random numbers it draws along the way. Accordingly, a probabilistic program has an associated *value function* which maps traces to output values. But the inference procedure relies on another function on traces, commonly called the *density*[1] of the program, which records a cumulative likelihood for the samples in a given trace. Approximating a normalised version of the density is the main challenge that inference algorithms aim to tackle. We will formalise these notions and demonstrate how the value function and density of a program are defined in terms of its operational semantics.

Gradient-Based Approximate Inference

Some of the most influential and practically important inference algorithms harness the gradient of the density functions they operate on, when these are differentiable. Generally the use of gradient-based techniques allows for much greater efficiency in inference.

A popular example is the Markov Chain Monte Carlo algorithm known as Hamiltonian Monte Carlo (HMC) [Duane et al., 1987, Neal, 2011]. Given a density function $g : X \to \mathbb{R}$, HMC samples are obtained as the states of a Markov chain by (approximately) simulating Hamilton's equations via an integrator that uses the gradient $\nabla_x g(x)$. Another important example is (stochastic) variational inference [Hoffman et al., 2013, Ranganath et al., 2014, Blei et al., 2017, Kucukelbir et al., 2015], which transforms the posterior inference problem into an optimisation problem.

In probabilistic programming, the above inference methods must be adapted to deal with the fact that in a universal PPL, the set of random primitives encountered can vary between executions, and traces can have arbitrary and unbounded dimension; moreover, the density function of a probabilistic program is generally not (everywhere) differentiable. Crucially, these adapted algorithms can only be applied when the derivative exists "often enough" since they at least have to *evaluate* the derivative. Thus, in the context of probabilistic programming, almost everywhere differentiability is often cited as a requirement for correctness [Zhou et al., 2019, Lee et al., 2020a]. Therefore, Hongseok Yang selected the following in his FSCD 2019 invited lecture [Yang, 2019] as one of three open problems in the field of semantics for probabilistic programs:

> *What class of statistical probabilistic programs have densities that are differentiable almost everywhere?*

In this thesis, we give an answer to this question and discuss ramifications for inference algorithms, in particular variational inference.

Variational Inference

In the variational inference approach to Bayesian statistics [Zhang et al., 2019, Murphy, 2012, Bishop, 2007, Blei et al., 2017], the problem of approximating difficult-to-compute posterior probability distributions is transformed into an optimisation problem. The idea is to approximate the posterior probability $p(\boldsymbol{z} \mid \boldsymbol{x})$ using a family $\mathcal{Q}$ of "simpler" densities $q(\boldsymbol{z})$ over the latent variables $\boldsymbol{z}$. The optimisation problem is then to find the member $q \in \mathcal{Q}$, which is "closest" to the true posterior $p(\boldsymbol{z} \mid \boldsymbol{x})$. In practice, variational inference has proven to yield good approximations much faster than MCMC.

[1] For some readers this terminology may be ambiguous; see Remark 2.4 for clarification.

Formally, the idea is captured by minimising the *KL-divergence* [Murphy, 2012, Bishop, 2007] between the variational approximation and the true posterior:

$$\arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\boldsymbol{z}) \parallel p(\boldsymbol{z} \mid \boldsymbol{x}))$$

Gradient-Based Optimisation

This task can be viewed as an instance of a more general class of optimisation problems, where we seek to find parameters $\boldsymbol{\theta}$ minimising (or maximising) an expectation:

$$\mathrm{argmin}_{\boldsymbol{\theta}} \, \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\theta}}(\boldsymbol{s})}[f(\boldsymbol{\theta}, \boldsymbol{s})]$$

In practice, variants of *stochastic gradient descent (SGD)* are frequently employed to solve such optimisation problems. In general, gradients of $\mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\theta}}(\boldsymbol{s})}[f(\boldsymbol{\theta}, \boldsymbol{s})]$ cannot be computed exactly. Therefore, Monte Carlo estimates $\widehat{g}_{\boldsymbol{\theta}_k}$ are employed as a surrogate.

In its simplest version, stochastic gradient descent follows Monte Carlo estimates of the gradient in each step:

$$\boldsymbol{\theta}_{k+1} \coloneqq \boldsymbol{\theta}_k - \gamma_k \cdot \widehat{g}_{\boldsymbol{\theta}_k} \qquad\qquad \widehat{g}_{\boldsymbol{\theta}_k} \sim \mathcal{G}_{\boldsymbol{\theta}_k}$$

where γ_k is the *step size* and $\mathcal{G}_{\boldsymbol{\theta}_k}$ is a distribution over the estimates.

For the correctness of SGD it is crucial that the estimation of the gradient is *unbiased*, i.e. correct in expectation:

$$\mathbb{E}_{\widehat{g}_{\boldsymbol{\theta}} \sim \mathcal{G}_{\boldsymbol{\theta}}}[\widehat{g}_{\boldsymbol{\theta}}] = \nabla_{\theta} \, \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\theta}}(\boldsymbol{z})}[f(\boldsymbol{\theta}, \boldsymbol{z})]$$

One such estimator is the *Score* or *REINFORCE* estimator [Ranganath et al., 2014, Wingate and Weber, 2013, Minh and Gregor, 2014, Mohamed et al., 2020].

Reparameterisation Gradient

Whilst the score estimator has the virtue of being very widely applicable, it unfortunately suffers from high variance, which can cause SGD to yield very poor results due to slow or unstable convergence.

The *reparameterisation gradient estimator*—the dominant approach in variational inference—instead reparameterises the latent variable $\boldsymbol{z}$ in terms of a base random variable $\boldsymbol{s}$ (viewed as the entropy source) via a diffeomorphic transformation. For example, if the distribution of the latent variable z is a Gaussian $\mathcal{N}(z \mid \mu, \sigma^2)$ with parameters $\boldsymbol{\theta} = \{\mu, \sigma\}$ then the location-scale transformation using the standard normal as the base distribution gives rise to the reparameterisation

$$z \sim \mathcal{N}(z \mid \mu, \sigma^2) \iff z = \varphi_{\mu,\sigma}(s), \quad s \sim \mathcal{N}(0, 1).$$

where $\varphi_{\mu,\sigma}(s) \coloneqq s \cdot \sigma + \mu$. The key advantage of this setup (often called "reparameterisation trick" [Kingma and Welling, 2014, Titsias and Lázaro-Gredilla, 2014, Rezende et al., 2014]) is that we have removed the dependency on $\boldsymbol{\theta}$ from the distribution w.r.t. which the expectation is taken (resulting in the distribution q). Therefore, we can now differentiate with respect to the parameters $\boldsymbol{\theta}$ of the variational distributions, succinctly

$$\nabla_{\boldsymbol{\theta}} \, \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\theta}}(\boldsymbol{z})}[f(\boldsymbol{\theta}, \boldsymbol{z})] = \nabla_{\boldsymbol{\theta}} \, \mathbb{E}_{\boldsymbol{s} \sim q(\boldsymbol{s})}[f(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))] = \mathbb{E}_{\boldsymbol{s} \sim q(\boldsymbol{s})}[\nabla_{\boldsymbol{\theta}} \, f(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))]$$

and the expectation on the right can be estimated with the Monte Carlo method.

The main benefit of the reparameterisation gradient estimator is that it has a significantly lower variance than the score estimator, resulting in faster convergence in practice.

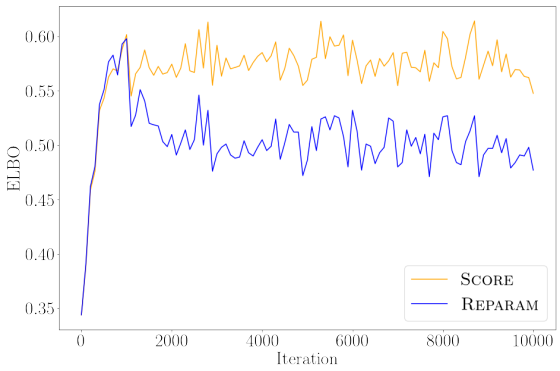


Figure 1.2: ELBO trajectories (higher means better) for Example 2.16 obtained with our implementation (cf. Chapter 7)

Bias of the Reparameterisation Gradient

Unfortunately, the reparameterisation gradient estimator is biased for non-differentiable models [Lee et al., 2018], which are readily expressible in programming languages with conditionals. Indeed, a single one-dimensional conditional encoding the (discontinuous) Heaviside[2] step function is sufficient to exhibit bias.

Crucially, *this may compromise correctness* of stochastic gradient descent and may yield poor results for the optimisation problem (cf. Fig. 2.5b).

1.2 Outline and Main Contributions

The high-level contribution of this thesis is to strengthen the foundation for *fast yet correct* Bayesian inference for probabilistic programming. The main ideas from Chapters 3 to 5 have been published in [Mak et al., 2021, Khajwal et al., 2023].

We provide an overview of the individual chapters and their contributions:

Chapter 2: Background

As an idealised statistical probabilistic programming language, we present statistical PCF (SPCF), an extension of call-by-value PCF with real numbers, primitive functions and constructs for sampling and conditioning. Furthermore, we review a variety of background material. In particular, we provide a concise introduction to variational inference, stochastic optimisation and gradient estimation.

Chapter 3: Densities of Almost Surely Terminating Programs are Differentiable Almost Everywhere

We provide an answer to Hongseok Yang's aforementioned question [Yang, 2019]: we demonstrate that both the density and value function are *differentiable almost everywhere* (that is, everywhere but on a set of measure zero), provided the program is *almost surely terminating*. Our result holds for statistical PCF provided the primitive functions (e.g. the set of analytic functions) satisfy mild closure properties.

[2] i.e. the characteristic function of the non-negative reals

We emphasise that it follows immediately that *purely deterministic programs with real parameters* denote functions that are almost everywhere differentiable.

Points of non-differentiability exist largely because of *branching*, which typically arises in a program when the control flow reaches a conditional statement. Hence, our work is a study of the connections between the traces of a probabilistic program and its branching structure. To achieve this we introduce *stochastic symbolic execution*, a form of operational semantics for probabilistic programs, designed to identify sets of traces corresponding to the same control-flow branch.

In the remaining chapters of this thesis we focus on variational inference and stochastic optimisation: we discuss fast and correct gradient-based optimisation methods via the reparameterisation estimator in the presence of conditionals, which compromise differentiability.

Chapter 4: PL Framework for Stochastic Optimisation

As the basis for our discussion, we present a programming language framework to pose stochastic optimisation problems generalising variational inference. We employ trace types to capture precisely the samples drawn in a fully eager call-by-value evaluation strategy and endow our language with a denotational semantics. To ensure that the optimisation problem is well-defined we discuss assumptions on primitive operations and present a type system $\vdash_{\text{int}}$ guaranteeing integrability.

Chapter 5: Correctness of Stochastic Gradient Descent with the Reparameterisation Gradient Estimator via Smoothing

Motivated by the biasedness of the reparameterisation gradient estimator for non-differentiable models, we propose a smoothed interpretation of our programming language, which is parameterised by an accuracy coefficient.

For terms typable in the $\vdash_{\text{int}}$-type system, we formally show that not only is the reparameterisation gradient estimator unbiased but also that stochastic gradient descent is correct. We analyse the relation between the smoothed (approximate) and the original objective function, and we propose another type system guaranteeing *uniform* convergence as the accuracy is enhanced. Thus, our smoothing approach in principle yields correct solutions up to arbitrary error tolerances by choosing a suitable accuracy coefficient and running stochastic gradient descent (with the reparameterisation estimator) on the smoothing.

Chapter 6: Diagonalisation Stochastic Gradient Descent

We propose a novel variant of SGD, *Diagonalisation Stochastic Gradient Descent* (DSGD), which avoids the choice of a fixed accuracy coefficient before running SGD. Instead, DSGD takes gradient steps of a *smoothed* model whilst simultaneously enhancing the accuracy of the approximation in each iteration.

We prove that asymptotically the *original*, unsmoothed optimisation problem is solved. In particular, we establish bounds on the variance of the estimator solely based on the syntactic structure of terms, which is required for the choice of a provably suitable step size scheme.

In summary, in contrast to other works, we verify the entire optimisation pipeline (incl. the well-definedness of the objective function and the correctness of SGD) and not just the unbiasedness of gradient estimation.

Chapter 7: Empirical Evaluation

We conduct an empirical evaluation demonstrating that our approach exhibits a similar convergence to an unbiased correction of the reparameterised gradient estimator by [Lee et al., 2018]—our main baseline. However, our estimator is simpler and more efficient: it is faster and attains orders of magnitude reduction in work-normalised variance. Besides, Diagonalisation Stochastic Gradient Descent exhibits a more stable convergence than using a standard stochastic gradient descent for *fixed*-accuracy smoothing.

Chapter 8: Continuous but Non-Differentiable Models

We prove that the reparameterisation gradient estimator is unbiased for continuous but non-differentiable models, and we investigate methods based on higher-order Horn logic with a background theory to verify continuity of terms with conditionals. We present a novel background theory for which the satisfiability problem is decidable by a new randomised algorithm. Furthermore, we present a reduction of continuity to a (decidable) satisfiability problem.

Finally, we conclude in Chapter 9 and propose future directions.

Chapter 2

Background

In this chapter we cover background material, starting with a brief review of measures and densities. Then we present an idealised higher-order statistical probabilistic programming language (Section 2.2). In Section 2.3, we provide an introduction to variational inference, which is followed by an account of stochastic optimisation and gradient estimation in Section 2.4. Thereafter, we review Schwartz densities and polynomially bounded functions (Section 2.5), analytic functions (Section 2.6) and notions of convergence for sequences of functions (Section 2.7).

Conventions. In the rest of the thesis we use bold font to represent sequences or vectors. For instance, we abbreviate the sequence of variables $x_1, \ldots, x_n$ to $\boldsymbol{x}$. Furthermore, we use shading (e.g. $x_1 \neq x_2$) to highlight noteworthy items. Moreover, further conventions are listed at the very end of this thesis, and an index is provided.

2.1 Measures and Densities

A ***measurable space*** is a pair (X, Σ_X) consisting of a set together with a *σ-algebra* of subsets, i.e. $\Sigma_X \subseteq \mathcal{P}(X)$ contains $\emptyset$ and is closed under complements and countable unions and intersections. Elements of Σ_X are called *measurable sets*. A ***measure*** on (X, Σ_X) is a function $\mu : \Sigma_X \to [0, \infty]$ satisfying $\mu(\emptyset) = 0$, and $\mu(\bigcup_{i \in I} U_i) = \sum_{i \in I} \mu(U_i)$ for every countable family $\{U_i\}_{i \in I}$ of pairwise disjoint measurable subsets. A measure μ is *finite* if $\mu(X) < \infty$. Finite measures are ***continuous from above***: if $(U_k)_{k \in \mathbb{N}}$ is a non-increasing sequence of measurable sets (i.e. $U_1 \supseteq U_2 \supseteq \cdots$) then $\mu(\bigcap_{k \in \mathbb{N}} U_k) = \lim_{k \to \infty} \mu(U_k)$. A (possibly partial) function $X \rightharpoonup Y$ is ***measurable*** if for every $U \in \Sigma_Y$ we have $f^{-1}(U) \in \Sigma_X$.

The space $\mathbb{R}$ of real numbers is an important example. The (Borel) σ-algebra $\Sigma_{\mathbb{R}}$ is the smallest one containing all intervals $[a, b)$, and the ***Lebesgue measure*** Leb is the unique measure on $(\mathbb{R}, \Sigma_{\mathbb{R}})$ satisfying $\mathrm{Leb}([a, b)) = b - a$. For measurable spaces (X, Σ_X) and (Y, Σ_Y), the *product σ-algebra* $\Sigma_{X \times Y}$ is the smallest one containing all $U \times V$, where $U \in \Sigma_X$ and $V \in \Sigma_Y$. So in particular we get for each $n \in \mathbb{N}$ a space $(\mathbb{R}^n, \Sigma_{\mathbb{R}^n})$, and additionally there is a unique measure Leb_n on $\mathbb{R}^n$ satisfying $\mathrm{Leb}_n(\prod_i U_i) = \prod_i \mathrm{Leb}(U_i)$.

When a function $f : X \to \mathbb{R}$ is measurable and μ is a measure on X, for each $U \in \Sigma_X$ we can define the *integral* $\int_U (\mathrm{d}\mu) f \in [0, \infty]$. If $X \subseteq \mathbb{R}^n$ is measurable we will also use the common notation $\int_U f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$ for $\int_U (\mathrm{d}\,\mathrm{Leb}_n) f$.

Table 2.1: Distributions and their pdfs

distribution	pdf	support	parameters
uniform	$\frac{[x\in(a,b)]}{b-a}$	(a,b)	$a<b\in\mathbb{R}$
normal	$\frac{1}{\sigma\sqrt{2\pi}}\exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$	$\mathbb{R}$	$\mu\in\mathbb{R},\sigma\in\mathbb{R}_{>0}$
half normal	$\frac{\sqrt{2}}{\sigma\sqrt{\pi}}\exp\left(\frac{-x^2}{2\sigma^2}\right)$	$\mathbb{R}_{\geq 0}$	$\sigma\in\mathbb{R}_{>0}$
exponential	$\lambda\cdot\exp(-\lambda x)$	$\mathbb{R}_{\geq 0}$	$\lambda\in\mathbb{R}_{>0}$
logistic	$\frac{\exp\left(-\frac{x-\mu}{s}\right)}{s\left(1+\exp\left(-\frac{x-\mu}{s}\right)\right)^2}$	$\mathbb{R}$	$\mu\in\mathbb{R},s\in\mathbb{R}_{>0}$
gamma	$\frac{1}{\Gamma(k)\theta^k}x^{k-1}\exp\left(-\frac{x}{\theta}\right)$	$\mathbb{R}_{>0}$	$k,\theta\in\mathbb{R}_{>0}$
beta	$x^{\alpha-1}\cdot(1-x)^{\beta-1}\cdot\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\cdot\Gamma(\beta)}$	$[0,1]$	$\alpha,\beta\in\mathbb{R}_{>0}$
Cauchy	$\left(\pi\gamma\left(1+\left(\frac{x-x_0}{\gamma}\right)^2\right)\right)^{-1}$	$\mathbb{R}$	$x_0\in\mathbb{R},\ \gamma\in\mathbb{R}_{>0}$

Common families of probability distributions on the reals (Uniform, Normal, etc., see Table 2.1) are examples of measures on $(\mathbb{R},\Sigma_{\mathbb{R}})$. We use the notation $\mathcal{N}(\mu,\sigma^2)$ for the normal distribution with parameters μ, σ^2, $\mathcal{N}$ for the standard normal distribution $\mathcal{N}(0,1)$ and $\mathcal{U}$ for the uniform distribution on $(0,1)$. Most often these are defined in terms of ***probability density functions*** with respect to the Lebesgue measure, meaning that for each $\mu_{\mathcal{D}}$ there is a measurable function $\mathrm{pdf}_{\mathcal{D}}:\mathbb{R}\to\mathbb{R}_{\geq 0}$ which determines it: $\mu_{\mathcal{D}}(U)=\int_U(\mathrm{d}\,\mathrm{Leb})\ \mathrm{pdf}_{\mathcal{D}}$, and the (measurable) set on which the density is positive is called its ***support*** (notation: supp). As we will see, density functions such as $\mathrm{pdf}_{\mathcal{D}}$ have a central place in Bayesian inference. Slightly abusing notation, we will sometimes refer to $\mathrm{pdf}_{\mathcal{D}}$ as $\mathcal{D}$.

Formally, if μ is a measure on a measurable space X, a ***density*** for μ with respect to another measure ν on X (most often ν is the Lebesgue measure) is a measurable function $f:X\to\mathbb{R}$ such that $\mu(U)=\int_U(\mathrm{d}\nu)f$ for every $U\in\Sigma_X$. In the context of the present work, an *inference algorithm* can be understood as a method for approximating a distribution of which we only know the density up to a normalising constant. In other words, if the algorithm is fed a (measurable) function $g:X\to\mathbb{R}$, it should produce samples approximating the probability measure $U\mapsto\frac{\int_U(\mathrm{d}\nu)g}{\int_X(\mathrm{d}\nu)g}$ on X.

Furthermore, if $U\subseteq\mathbb{R}^n$ is measurable, $\boldsymbol{\mathcal{D}}:U\to\mathbb{R}_{\geq 0}$ is a probability density function and $f:\mathbb{R}^n\to\mathbb{R}$ is measurable we will write $\mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\mathcal{D}}}[f(\boldsymbol{x})]$ for $\int_U\boldsymbol{\mathcal{D}}(\boldsymbol{x})\cdot f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$.

We will make use of some basic notions from topology: given a topological space X and a set $A\subseteq X$, the ***interior*** of A is the largest open set $\mathring{A}$ contained in A. Dually the ***closure*** of A is the smallest closed set $\overline{A}$ containing A, and the ***boundary*** of A is defined as $\partial A\coloneqq\overline{A}\setminus\mathring{A}$. Note that for all $U\subseteq\mathbb{R}^n$, all of $\mathring{U}$, $\overline{U}$ and ∂U are measurable (in $\Sigma_{\mathbb{R}^n}$).

2.2 A Statistical Probabilistic Programming Language

In this section, we introduce the syntax and (operational) semantics of our idealised statistical probabilistic programming language.

Statistical PCF (SPCF) is a higher-order probabilistic programming language with recursion in purified form. It is a statistical probabilistic version of call-by-value PCF [Scott, 1993, Sieber, 1990] with reals as the ground type and an (inconsequential) variant of CBV SPCF [Vákár et al., 2019] and a (CBV) extension of PPCF [Ehrhard et al., 2018] with scoring; it may be viewed as a simply-typed version of the untyped probabilistic languages of [Borgström et al., 2016, Culpepper and Cobb, 2017, Wand et al., 2018].

The execution of a probabilistic program generates a *trace*: a sequence containing the values sampled during a run. Our operational semantics captures this dynamic perspective. This is closely related to the treatment in [Borgström et al., 2016] which, following [Kozen, 1979], views a probabilistic program as a deterministic program parameterised by the sequence of random draws made during the evaluation.

2.2.1 Syntax

The ***raw terms*** of our programming language are defined by the grammar:

$$\begin{aligned} M ::= {} & x \mid \underline{f}(M,\dots,M) \mid \mathbf{if}\, M < 0\, \mathbf{then}\, M\, \mathbf{else}\, M \\ & \mid \lambda x.\, M \mid M\, M \mid \mathbf{Y} M \\ & \mid \mathbf{sample}_{\mathcal{D}} \mid \mathbf{score}(M) \end{aligned}$$

where x ranges over *variables* (taken from a denumerable collection), $f \in \mathrm{Op}$ are ***primitive operations***, and $\mathcal{D} \in \mathrm{Dist}$ is a probability density function over $\mathbb{R}$ (potentially with a support which is a strict subset of $\mathbb{R}$, see Table 2.1). Recursion is provided by the standard fixed point combinator $\mathbf{Y}$. The probabilistic constructs of SPCF are relatively standard (see for example [Staton, 2017]).

The sampling construct $\mathbf{sample}_{\mathcal{D}}$ draws a sample from $\mathcal{D}$. For instance $\mathbf{sample}_{\mathcal{N}(0,1)}$ draws a sample from the standard normal distribution.

The **score** construct enables conditioning on observed data by multiplying the weight of the current execution with the (non-negative) real number denoted by M.

We impose various constraints on the set of primitive operations Op and distributions Dist throughout this thesis. Constants can be viewed as 0-ary functions and will be abbreviated as $\underline{r}$ (for $r \in \mathbb{R}$). Besides, Op may include standard arithmetic operations for which we use the customary infix, postfix and prefix notation: $M \mathbin{\underline{+}} N$ (addition), $M \mathbin{\underline{\cdot}} N$ (multiplication), $M\underline{^{-1}}$ (inverse), $\underline{-}M$ (numeric negation), $\underline{\exp}\, M$ (exponentiation), and $\underline{\log}\, M$ (natural logarithm). We frequently omit the underline to reduce clutter.

Example 2.1. $(\lambda x.\, \mathbf{score}(\mathbf{if}\, x - \mathbf{sample}_{\mathcal{U}} < 0\, \mathbf{then}\, 0\, \mathbf{else}\, x))\, \mathbf{sample}_{\mathcal{U}}$ represents the term $(\lambda x.\, \mathbf{score}(\mathbf{if}\, \underline{-}(x, \mathbf{sample}_{\mathcal{U}}) < 0\, \mathbf{then}\, \underline{0}\, \mathbf{else}\, x))\, \mathbf{sample}_{\mathcal{U}}$, where "$-$" is the (binary) subtraction function and $\mathcal{U}$ is the uniform distribution on $(0,1)$.

Simple Type System

(Simple) types are generated from a base type R representing real numbers

$$\tau ::= R \mid \tau \to \tau$$

$$\frac{}{\Gamma, x : \tau \vdash x : \tau} \qquad \frac{\Gamma \vdash M_1 : R \quad \cdots \quad \Gamma \vdash M_\ell : R}{\Gamma \vdash \underline{f}(M_1, \ldots, M_\ell) : R}\ f : \mathbb{R}^\ell \rightharpoonup \mathbb{R} \in \mathrm{Op}$$

$$\frac{\Gamma \vdash L : R \quad \Gamma \vdash M : \tau \quad \Gamma \vdash N : \tau}{\Gamma \vdash \mathbf{if}\, L < 0\, \mathbf{then}\, M\, \mathbf{else}\, N : \tau}$$

$$\frac{\Gamma, x : \tau_1 \vdash M : \tau_2}{\Gamma \vdash \lambda x.\, M : \tau_1 \to \tau_2} \qquad \frac{\Gamma \vdash M : \tau_1 \to \tau_2 \quad \Gamma \vdash N : \tau_1}{\Gamma \vdash M\, N : \tau_2}$$

$$\frac{\Gamma \vdash M : (\tau_1 \to \tau_2) \to (\tau_1 \to \tau_2)}{\Gamma \vdash \mathbf{Y} M : \tau_1 \to \tau_2}$$

$$\frac{}{\Gamma \vdash \mathbf{sample}_{\mathcal{D}} : R} \qquad \frac{\Gamma \vdash M : R}{\Gamma \vdash \mathbf{score}(M) : R}$$

Figure 2.1: Typing judgements of SPFC.

A context (typically denoted by Γ) is a finite list of typed variables $x_1 : \tau_1, \ldots, x_m : \tau_m$; typing judgements are standard and presented in Fig. 2.1.

Finally, we write Λ for the set of (well-typed) SPCF terms, and Λ^0 for the set of closed SPCF terms.

In the interest of readability, we sometimes use pseudocode (e.g. Fig. 1.1a) in the style of Core ML to express SPCF terms.

Example 2.2 (Running Example Ped)**.** We express in SPCF the example in Fig. 1.1a.

$$\mathrm{Ped} \equiv \begin{pmatrix} \mathbf{let}\ x = \mathbf{sample}_{\mathcal{U}} \cdot \underline{3}\ \mathbf{in} \\ \mathbf{let}\ d = \mathrm{walk}\, x\ \mathbf{in} \\ \mathbf{let}\ w = \mathbf{score}(\underline{\mathrm{pdf}_{\mathcal{N}(1.1,0.1)}}(d))\ \mathbf{in}\ x \end{pmatrix} \qquad \text{where}$$

$$\mathrm{walk} \equiv \mathbf{Y} \begin{pmatrix} \lambda f x. & \mathbf{if}\ x \leq 0\ \mathbf{then}\ \underline{0} \\ & \mathbf{else}\ \begin{pmatrix} \mathbf{let}\ s = \mathbf{sample}_{\mathcal{U}}\ \mathbf{in} \\ \mathbf{if}\, \mathbf{sample}_{\mathcal{U}} < 0.5\, \mathbf{then}\, \big(s + f(x+s)\big)\, \mathbf{else}\, \big(s + f(x-s)\big) \end{pmatrix} \end{pmatrix}$$

The let construct, $\mathbf{let}\ x = N\ \mathbf{in}\ M$, is syntactic sugar for the term $(\lambda x.M)\, N$; and $\mathrm{pdf}_{\mathcal{N}(1.1,0.1)}$, the density function of the normal distribution with mean 1.1 and variance 0.1, is a primitive function.

2.2.2 Operational Sampling Semantics

To approximate the posterior, inference engines for probabilistic programs often proceed indirectly and operate on the space of *program traces*, rather than on the space of possible return values. By *trace*, we mean the sequence of samples drawn in the course of a particular run, one for each random primitive encountered. Because each random primitive (qua probability distribution) in the language comes with a density, given a particular trace we can compute a coefficient as the appropriate product. We can then multiply this coefficient by all scores encountered in the execution, and this yields a (*weight*) function, mapping traces to the non-negative reals, over which the chosen inference algorithm may operate. This indirect approach is more practical, and enough to answer the query, since every trace unambiguously induces a return value.

Traces

Recall that in our language, $\mathbf{sample}_{\mathcal{D}}$ produces a random value in R; accordingly a ***trace*** is a finite sequence of reals. (In Chapter 3, we will restrict distributions to the uniform distribution on $(0,1)$ and traces are sequences of elements $(0,1)$ accordingly.) We define a *measure space* $\mathbb{S}$ *of traces* to be the set $\bigcup_{n\in\mathbb{N}} \mathbb{R}^n$, equipped with the standard disjoint union σ-algebra, and the sum of the respective (higher-dimensional) Lebesgue measures. Formally, writing $\mathbb{S}_n \coloneqq \mathbb{R}^n$, we define:

$$\mathbb{S} \coloneqq \left(\bigcup_{n\in\mathbb{N}} \mathbb{S}_n, \left\{\bigcup_{n\in\mathbb{N}} U_n \mid U_n \in \Sigma_{\mathbb{S}_n}\right\}, \mu_{\mathbb{S}}\right) \text{ and } \mu_{\mathbb{S}}\left(\bigcup_{n\in\mathbb{N}} U_n\right) \coloneqq \sum_{n\in\mathbb{N}} \mathrm{Leb}_n(U_n).$$

Henceforth, we write traces as lists, such as $[0.5, 0.999, 0.12]$; the empty trace as $[]$; and the concatenation of traces $\boldsymbol{s}, \boldsymbol{s}' \in \mathbb{S}$ as $\boldsymbol{s} +\!\!+ \boldsymbol{s}'$.

More generally, to account for open terms, we define, for each $m \in \mathbb{N}$, the measure space

$$\mathbb{R}^m \times \mathbb{S} \coloneqq \left(\bigcup_{n\in\mathbb{N}} \mathbb{R}^m \times \mathbb{S}_n, \left\{\bigcup_{n\in\mathbb{N}} V_n \mid V_n \in \Sigma_{\mathbb{R}^m\times\mathbb{S}_n}\right\}, \mu_{\mathbb{R}^m\times\mathbb{S}}\right)$$

where $\mu_{\mathbb{R}^m\times\mathbb{S}}\left(\bigcup_{n\in\mathbb{N}} V_n\right) \coloneqq \sum_{n\in\mathbb{N}} \mathrm{Leb}_{m+n}(V_n)$. To avoid clutter, we will elide the subscript from $\mu_{\mathbb{R}^m\times\mathbb{S}}$ whenever it is clear from the context.

Small-Step Reduction

Next, we define the ***values*** (typically denoted V), ***redexes*** (typically R) and ***evaluation contexts*** (typically E):

values	$V ::= \underline{r} \mid \lambda y.M$
redexes	$R ::= \underline{f}(\underline{r_1}, \ldots, \underline{r_\ell}) \mid \mathbf{if}\ \underline{r} < 0\ \mathbf{then}\ M\ \mathbf{else}\ N$
	$\mid \mathbf{Y}(\lambda y.\, M) \mid (\lambda y.\, M)\, V \mid$
	$\mid \mathbf{sample}_{\mathcal{D}} \mid \mathbf{score}(\underline{r})$
evaluation contexts	$E ::= [] \mid \underline{f}(\underline{r_1}, \ldots, \underline{r_{i-1}}, E, M_{i+1}, \ldots, M_\ell)$
	$\mid \mathbf{if}\ E < 0\ \mathbf{then}\ M\ \mathbf{else}\ N$
	$\mid E\, M \mid (\lambda y.M)\, E \mid \mathbf{Y}E$
	$\mid \mathbf{score}(E)$

We write Λ_v for the set of SPCF values, and Λ_v^0 for the set of closed SPCF values.

It is easy to see that every closed SPCF term M is either a value, or there exists a unique pair of context E and redex R such that $M \equiv E[R]$.

We now present the operational semantics of SPCF as a rewrite system of ***configurations***, which are triples of the form $\langle M, w, \boldsymbol{s}\rangle$ where M is a closed SPCF term, $w \in \mathbb{R}_{\geq 0}$ is a *weight*, and $\boldsymbol{s} \in \mathbb{S}$ a trace. (We will sometimes refer to these as the *concrete* configurations, in contrast with the *abstract* configurations of our symbolic operational semantics, see Section 3.2.2.)

The small-step reduction relation $\to$ is defined in Fig. 2.2. In the rule for $\mathbf{sample}_{\mathcal{D}}$, a random value $r \in \mathbb{R}$ is generated and recorded in the trace, while the weight is multiplied with the probability density function of $\mathcal{D}$ evaluated at r. In the rule for $\mathbf{score}(\underline{r})$, the

Redex Contractions:

$$\langle \underline{f}(\underline{r_1}, \ldots, \underline{r_\ell}), w, \boldsymbol{s} \rangle \to \left\langle \underline{f(r_1, \ldots, r_\ell)}, w, \boldsymbol{s} \right\rangle \qquad (\text{if } (r_1, \ldots, r_\ell) \in \operatorname{dom} f)$$

$$\langle \underline{f}(\underline{r_1}, \ldots, \underline{r_\ell}), w, \boldsymbol{s} \rangle \to \text{fail} \qquad (\text{if } (r_1, \ldots, r_\ell) \notin \operatorname{dom} f)$$

$$\langle \textbf{if } \underline{r} < 0 \textbf{ then } M \textbf{ else } N, w, \boldsymbol{s} \rangle \to \langle M, w, \boldsymbol{s} \rangle \qquad (\text{if } r < 0)$$

$$\langle \textbf{if } \underline{r} < 0 \textbf{ then } M \textbf{ else } N, w, \boldsymbol{s} \rangle \to \langle N, w, \boldsymbol{s} \rangle \qquad (\text{if } r \geq 0)$$

$$\langle (\lambda y.\, M)\, V, w, \boldsymbol{s} \rangle \to \langle M[V/y], w, \boldsymbol{s} \rangle$$

$$\langle \mathbf{Y}(\lambda y.M), w, \boldsymbol{s} \rangle \to \langle \lambda z.M[\mathbf{Y}(\lambda y.M)/y]\, z, w, \boldsymbol{s} \rangle \qquad (\text{for fresh variable } z)$$

$$\langle \textbf{sample}_{\mathcal{D}}, w, \boldsymbol{s} \rangle \to \left\langle \underline{r}, \; w \cdot \mathcal{D}(r) \, , \; \boldsymbol{s} +\!\!+ [r] \right\rangle \qquad (\text{for } r \in \mathbb{R})$$

$$\langle \textbf{score}(\underline{r}), w, \boldsymbol{s} \rangle \to \langle \underline{r}, \; r \cdot w \, , \boldsymbol{s} \rangle \qquad (\text{if } r \geq 0)$$

$$\langle \textbf{score}(\underline{r}), w, \boldsymbol{s} \rangle \to \text{fail} \qquad (\text{if } r < 0)$$

Evaluation Contexts:

$$\frac{\langle R, w, \boldsymbol{s} \rangle \to \langle R', w', \boldsymbol{s}' \rangle}{\langle E[R], w, \boldsymbol{s} \rangle \to \langle E[R'], w', \boldsymbol{s}' \rangle} \qquad \frac{\langle R, w, \boldsymbol{s} \rangle \to \text{fail}}{\langle E[R], w, \boldsymbol{s} \rangle \to \text{fail}}$$

Figure 2.2: Operational small-step semantics of SPCF

current weight is multiplied by non-negative $r \in \mathbb{R}$: typically this reflects the likelihood of the current execution given some observed data. Similarly to [Borgström et al., 2016] we reduce terms which cannot be reduced in a reasonable way (i.e. scoring with negative constants or evaluating functions outside their domain) to fail.

Example 2.3. We present a possible reduction sequence for the program in Example 2.2:

$$\begin{aligned}
\langle \mathsf{Ped}, 1, [] \rangle &\to^* \left\langle \begin{pmatrix} \textbf{let } x = \underline{0.2} \cdot \underline{3} \textbf{ in} \\ \textbf{let } d = \text{walk } x \textbf{ in} \\ \textbf{let } w = \textbf{score}(\underline{\text{pdf}_{\mathcal{N}(1.1,0.1)}}(d)) \textbf{ in } x \end{pmatrix}, 1, [0.2] \right\rangle \\
&\to^* \left\langle \begin{pmatrix} \textbf{let } d = \text{walk } \underline{0.6} \textbf{ in} \\ \textbf{let } w = \textbf{score}(\underline{\text{pdf}_{\mathcal{N}(1.1,0.1)}}(d)) \textbf{ in } \underline{0.6} \end{pmatrix}, 1, [0.2] \right\rangle \\
&\to^* \left\langle \textbf{let } w = \textbf{score}(\underline{\text{pdf}_{\mathcal{N}(1.1,0.1)}}(\underline{0.9})) \textbf{ in } \underline{0.6}, 1, [0.2, 0.9, 0.7] \right\rangle \qquad (\star) \\
&\to^* \langle \textbf{let } w = \textbf{score}(\underline{0.54}) \textbf{ in } \underline{0.6}, 1, [0.2, 0.9, 0.7] \rangle \\
&\to^* \langle \underline{0.6}, 0.54, [0.2, 0.9, 0.7] \rangle
\end{aligned}$$

In this execution, the initial $\textbf{sample}_{\mathcal{U}}$ yields $\underline{0.2}$, which is appended to the trace. At step $(\star)$, we assume given a reduction sequence $\langle \text{walk } \underline{0.6}, 1, [0.2] \rangle \to^* \langle \underline{0.9}, 1, [0.2, 0.9, 0.7] \rangle$; this means that in the call to walk, 0.9 was sampled as the step size and 0.7 as the direction factor; this makes the new location -0.3, which is negative, so the return value is 0.9. In the final step, we perform *conditioning* using the likelihood of observing 0.9 given the data 1.1: the **score** expression updates the current weight using the density of 0.9 in the normal distribution with parameters $(1.1, 0.1)$.

2.2.3 Value and Weight Functions

Using the relation $\to$, we now aim to reason more globally about probabilistic programs in terms of the traces they produce. Let M be an SPCF term with free variables amongst $x_1, \dots, x_m$ of type R. Its ***value function*** $\mathrm{value}_M : \mathbb{R}^m \times \mathbb{S} \to \Lambda_v^0 \cup \{\bot\}$ returns, given values for each free variable and a trace, the output value of the program, if the program terminates in a value. The ***weight function*** $\mathrm{weight}_M : \mathbb{R}^m \times \mathbb{S} \to \mathbb{R}_{\geq 0}$ returns the final weight of the corresponding execution. Formally:

$$\mathrm{value}_M(\boldsymbol{r}, \boldsymbol{s}) \coloneqq \begin{cases} V & \text{if } \langle M[\underline{\boldsymbol{r}}/\boldsymbol{x}], 1, [] \rangle \to^* \langle V, w, \boldsymbol{s} \rangle \\ \bot & \text{otherwise} \end{cases}$$

$$\mathrm{weight}_M(\boldsymbol{r}, \boldsymbol{s}) \coloneqq \begin{cases} w & \text{if } \langle M[\underline{\boldsymbol{r}}/\boldsymbol{x}], 1, [] \rangle \to^* \langle V, w, \boldsymbol{s} \rangle \\ 0 & \text{otherwise} \end{cases}$$

For closed SPCF terms M we just write $\mathrm{weight}_M(\boldsymbol{s})$ for $\mathrm{weight}_M([], \boldsymbol{s})$ (similarly for value_M), and it follows already from [Borgström et al., 2016, Lemma 9] that the functions value_M and weight_M are measurable if the primitive operations in Op are measurable (see also Section 3.1.1).

Finally, every closed SPCF term M has an associated ***value measure***

$$\mu_M : \Sigma_{\Lambda_v^0} \to \mathbb{R}_{\geq 0}$$

defined by $\mu_M(U) \coloneqq \int_{\mathrm{value}_M{}^{-1}(U)} \mathrm{d}\mu_{\mathbb{S}}\ \mathrm{weight}_M$. This corresponds to the denotational semantics of SPCF in the ω-quasi-Borel space model via computational adequacy [Vákár et al., 2019].

Remark 2.4. In much of the probabilistic programming literature (e.g. [Lee et al., 2020a, Zhou et al., 2019, Yang, 2019], including this thesis), the above-mentioned weight function on traces is called the *density* of the probabilistic program. This may be confusing: as we have seen, a probabilistic program induces a posterior probability distribution on return values, and it is natural to ask whether this distribution admits a probability density function (Radon-Nikodym derivative) w.r.t. some base measure. This problem is of current interest [Bhat et al., 2012, Bhat et al., 2017, Ismail and Shan, 2016] but unrelated to the present work.

To distinguish them, let's refer to the weight function of the program, weight_M, as its *trace density*, and the Radon-Nikodym derivative of the program's value-measure, $\frac{d\mu_M}{d\nu}$ where ν is the reference measure of the measurable space $\Sigma_{\Lambda_v^0}$, as the *output density*. Observe that, for any measurable function $f : \Lambda_v^0 \to [0, \infty]$,

$$\int_{\Lambda_v^0} d\,\mu_M\, f = \int_{\mathrm{value}_M^{-1}(\Lambda_v^0)} d\mu_{\mathbb{S}}\ \mathrm{weight}_M \cdot (f \circ \mathrm{value}_M) = \int_{\mathbb{S}} d\mu_{\mathbb{S}}\ \mathrm{weight}_M \cdot (f \circ \mathrm{value}_M)$$

(because if $\boldsymbol{s} \notin \mathrm{value}_M^{-1}(\Sigma_{\Lambda_v^0})$ then $\mathrm{weight}_M(\boldsymbol{s}) = 0$). It follows that we can express any expectation w.r.t. the output density $\frac{d\mu_M}{d\nu}$ as an expectation w.r.t. the trace density weight_M. If our aim is, instead, to generate samples from $\frac{d\mu_M}{d\nu}$ then we can simply generate samples from weight_M, and deterministically convert each sample to the space $(\Lambda_v^0, \Sigma_{\Lambda_v^0})$ via the value function value_M. In other words, if our intended output is just a sequence of samples, then our inference engine does not need to concern itself with the consequences of change of variables.

2.3 Variational Inference

Variational inference [Zhang et al., 2019, Murphy, 2012, Bishop, 2007, Blei et al., 2017] frames posterior inference as a (deterministic) optimisation problem and has proven in practice to yield good approximations faster than MCMC. The idea is to approximate the posterior probability $p(\boldsymbol{z} \mid \boldsymbol{x})$ using "simpler" densities $q_{\boldsymbol{\theta}}(\boldsymbol{z})$ over the latent variables $\boldsymbol{z}$, parameterised by $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^m$, from which we can sample easily (in contrast to the posterior). The $q_{\boldsymbol{\theta}}(\boldsymbol{z})$ are often referred to as ***variational approximations*** and $\mathcal{Q} \coloneqq \{q_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ as the ***variational family***. Then the aim is to find the member of the variational family which is "closest" to the actual posterior.

Example 2.5. Consider the probabilistic program

```
let z  = sample 𝒩(0,1)
    mu = if z < 0 then 0 else 1
    observe x from 𝒩(mu,1)
in  mu
```

which defines the joint density

$$p(z,x) \coloneqq \mathcal{N}(z \mid 0,1) \cdot \begin{cases} \mathcal{N}(x \mid 0,1) & \text{if } z < 0 \\ \mathcal{N}(x \mid 1,1) & \text{otherwise} \end{cases}$$

As the variational family we can choose the univariate normal distributions with mean θ and variance 1: $q_\theta(z) \coloneqq \mathcal{N}(z \mid \theta, 1)$. We plot some members of the variational family in Fig. 2.3b. For this simple joint distribution we can quite easily compute the posterior density exactly, which is governed by Bayes' theorem

$$p(z \mid x) = \frac{p(z,x)}{\underbrace{\int p(z,x)\,\mathrm{d}z}_{\text{normalising constant}}}$$

by computing the normalising constant

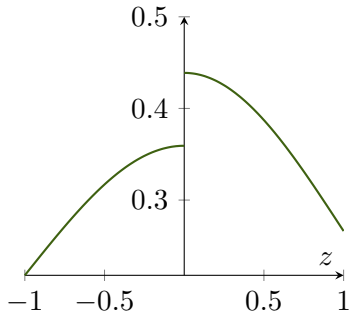
$$\begin{aligned} C_x &\coloneqq \int p(z,x)\,\mathrm{d}z \\ &= \mathcal{N}(x \mid 0,1) \cdot \int_{-\infty}^{0} \mathcal{N}(z \mid 0,1)\,\mathrm{d}z + \mathcal{N}(x \mid 1,1) \cdot \int_{0}^{\infty} \mathcal{N}(z \mid 0,1)\,\mathrm{d}z \\ &= \frac{\mathcal{N}(x \mid 0,1) + \mathcal{N}(x \mid 1,1)}{2} \end{aligned}$$

Thus, the posterior density is $p(z \mid x) = \frac{p(z,x)}{C_x}$ and we plot $p(z \mid x = 0.7)$ in Fig. 2.3a.

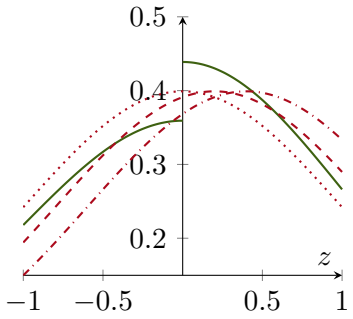
The intuitive notion of "closeness" is usually formalised by the ***Kullback–Leibler divergence (KL divergence)***:

$$\mathrm{KL}(q(\boldsymbol{z}) \parallel p(\boldsymbol{z})) \coloneqq \mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z})} \left[\log q(\boldsymbol{z}) - \log p(\boldsymbol{z})\right]$$

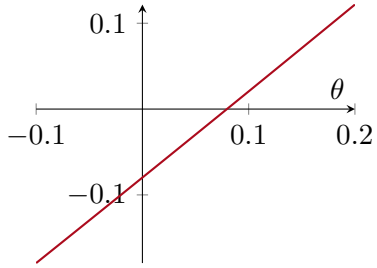
The KL divergence is only well-defined if q is *absolutely continuous* w.r.t. p: for almost all $\boldsymbol{z}$ such that $p(\boldsymbol{z}) = 0$, $q(\boldsymbol{z}) = 0$.



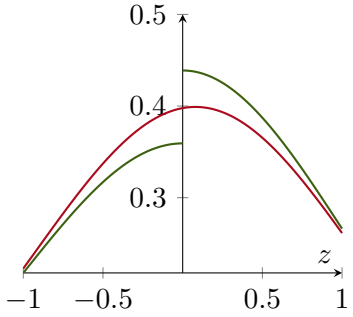
(a) Posterior density $p(z \mid x = 0.7)$



(b) Variational approximations (red, $\theta \in \{0, 0.2, 0.4\}$)



(c) Derivative of KL-divergence



(d) Optimal variational approximation ($\theta \approx 0.08$)

Figure 2.3: Posterior density and (optimal) variational approximations for Example 2.5

Intuitively, $\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}[-\log p(\boldsymbol{z})]$ rewards q which place mass where p is high, whilst $\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}[\log q(\boldsymbol{z})]$, which is also known as *entropy*, penalises q which are not spread out. The latter is crucial for otherwise all mass could be placed where the posterior has a maximum, whilst ignoring the rest.

Example 2.6. An elementary calculation reveals that for the posterior density and variational approximations of Example 2.5,

$$\frac{\partial}{\partial\theta}\,\mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{z}) \parallel p(\boldsymbol{z})) = \theta - 0.2 \cdot \mathcal{N}(\theta \mid 0, 1)$$

Therefore, the KL-divergence is minimised for $\theta \approx 0.08$, where the derivative vanishes (see Figs. 2.3c and 2.3d).

We recall some basic properties:

Lemma 2.7. *The KL-divergence is non-negative and* $\mathrm{KL}(q \parallel p) = 0$ *iff* $q = p$ *almost everywhere.*

Proof. By Jensen's inequality (applied to $-\log$),

$$\mathrm{KL}(q \parallel p) = \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}\left[-\log\frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}\right] \geq -\log\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}\left[\frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}\right] = -\log\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[1] = 0$$

Besides, if $\mathrm{KL}(q \parallel p) = 0$ then since $-\log$ is strictly convex, $\log\frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}$ must be constant (and equal to 0) a.e. and therefore $q(\boldsymbol{z}) = p(\boldsymbol{z})$ a.e. (The "if"-implication of the second claim is trivial.) □

Instantiating the posterior density $p(\boldsymbol{z} \mid \boldsymbol{x})$ for p, we can state the problem variational inference addresses:

Problem 2.8. Variational Inference

Given: Posterior distribution $p(\boldsymbol{z} \mid \boldsymbol{x})$, parameter space $\boldsymbol{\Theta} \subseteq \mathbb{R}^m$ and distributions $q_{\boldsymbol{\theta}}(\boldsymbol{z})$ with variational parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta}$

Find: $\arg\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{z}) \parallel p(\boldsymbol{z} \mid \boldsymbol{x}))$

A priori, it is not clear whether this problem is at all helpful because the whole point of Bayesian inference is to obtain the posterior and hence the objective function is generally intractable. However, we can note $p(\boldsymbol{z} \mid \boldsymbol{x}) = p(\boldsymbol{z}, \boldsymbol{x})/p(\boldsymbol{x})$ and massage the KL divergence:

$$\begin{aligned}
\mathrm{KL}(q(\boldsymbol{z}) \parallel p(\boldsymbol{z} \mid \boldsymbol{x})) &= \mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z})} \left[\log q(\boldsymbol{z}) - \log p(\boldsymbol{z} \mid \boldsymbol{x})\right] \\
&= \mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z})} \left[\log q(\boldsymbol{z}) - \log \left(\frac{p(\boldsymbol{z}, \boldsymbol{x})}{p(\boldsymbol{x})}\right)\right] \\
&= \mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z})} \left[\log q(\boldsymbol{z}) - \log p(\boldsymbol{z}, \boldsymbol{x})\right] + \log p(\boldsymbol{x})
\end{aligned}$$

Whilst $\log p(\boldsymbol{x})$ is still generally intractable, it is independent from q. Thus, minimising the KL divergence is equivalent to maximising the negation of the left summand, which is referred to as ***evidence lower bound (ELBO)***:

$$\mathrm{ELBO}(q) \coloneqq \mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z})} \left[\log p(\boldsymbol{z}, \boldsymbol{x}) - \log q(\boldsymbol{z})\right]$$

$p(\boldsymbol{x})$ is also known as *(model) evidence*. Thus, the terminology can be explained by noting

$$\log p(\boldsymbol{x}) = \mathrm{ELBO}(q) + \mathrm{KL}(q(\boldsymbol{z}) \parallel p(\boldsymbol{z} \mid \boldsymbol{x})) \geq \mathrm{ELBO}(q)$$

since the KL divergence is non-negative.

Consequently, we can re-phrase Problem 2.8 as follows:

Problem 2.9. Variational Inference (ELBO version)

Given: Joint distribution $p(\boldsymbol{z}, \boldsymbol{x})$, parameter space $\boldsymbol{\Theta} \subseteq \mathbb{R}^m$ and distributions $q_{\boldsymbol{\theta}}(\boldsymbol{z})$ with variational parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta}$

Find: $\arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\theta}}(\boldsymbol{z})} \left[\log p(\boldsymbol{z}, \boldsymbol{x}) - \log q_{\boldsymbol{\theta}}(\boldsymbol{z})\right]$

Of course, the choice of the variational family is important in practice: it needs to be sufficiently rich to offer good approximations to the posterior, but it also needs to satisfy the absolute continuity constraint for the KL-divergence to be well-defined. A choice popular in traditional statistics are independent univariate Gaussians, which is known as the *mean-field* variational family [Blei et al., 2017, Zhang et al., 2019, Opper and Saad, 2001]. Furthermore, special cases of both models and variational families (falling in *exponential families*) have been identified which admit closed form solutions [Hoffman et al., 2013, Blei et al., 2017].

In probabilistic programming systems the variational families can be either derived automatically from models or users can supply custom approximations expressed *within*

the language, which are often referred to as ***guides***. To ensure that, in contrast to the posterior, we can sample efficiently from the guide, it cannot contain observations. One of the benefits of probabilistic programming is that via automatic inference the paradigm allows practitioners to explore variational families rapidly.

Finally, note that alternative objective functions quantifying the "difference" between distributions are also conceivable; [Zhang et al., 2019] contains a survey of competing approaches.

2.4 Stochastic Optimisation and Gradient Estimation

More generally, we wish to solve optimisation problems of the following form, which not only covers variational inference but also has applications in fields ranging from reinforcement learning and queuing theory to portfolio design [Mohamed et al., 2020, Sutton and Barto, 2018]:

Problem 2.10. General Optimisation of Expectations

Given: Parameter space $\boldsymbol{\Theta} \subseteq \mathbb{R}^m$, probability density function $\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}} : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ for $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and measurable function $f : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$

Find: $\arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathbb{E}_{\boldsymbol{z} \sim \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})} [f(\boldsymbol{\theta}, \boldsymbol{z})]$

As in the rest of this thesis, we focus on continuous distributions.

Arguably the most popular optimisation method in practice are gradient methods, which are iterative methods harnessing gradient information. Unfortunately, gradients of the objective function of Problem 2.10, $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z} \sim \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})} [f(\boldsymbol{\theta}, \boldsymbol{z})]$, cannot be computed exactly in general due the presence of the expectation. As a surrogate, we *estimate* gradients $\widehat{g}_{\boldsymbol{\theta}}$ with Monte Carlo methods [Metropolis and Ulam, 1949], which entails drawing samples from a distribution $\boldsymbol{\mathcal{G}}_{\theta_k}$. In its simplest form, ***Stochastic Gradient Descent (SGD)*** then iteratively follows the estimates:

$$\boldsymbol{\theta}_{k+1} \coloneqq \boldsymbol{\theta}_k - \gamma_k \cdot \widehat{g}_{\boldsymbol{\theta}_k} \qquad\qquad \widehat{g}_{\boldsymbol{\theta}_k} \sim \boldsymbol{\mathcal{G}}_{\boldsymbol{\theta}_k} \tag{SGD}$$

where γ_k is the ***step size***.

When applying the gradient estimators (e.g. in SGD), it is usually crucial that they are correct in expectation, a property which is known as ***unbiasedness***:

$$\mathbb{E}_{\widehat{g}_{\boldsymbol{\theta}} \sim \boldsymbol{\mathcal{G}}_{\boldsymbol{\theta}}} [\widehat{g}_{\boldsymbol{\theta}}] = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z} \sim \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})} [f(\boldsymbol{\theta}, \boldsymbol{z})] \tag{2.1}$$

Intuitively, this ensures that in expectation SGD moves into the correct direction.

Furthermore, the estimator, which is a random variable, should be as "close" to the actual gradient as possible. Intuitively, this ensures that it is not likely that we go into the "wrong" direction too often. We can formalise this desideratum by aspiring the variance to be as low as possible, which is

$$\mathbb{E}_{\widehat{g}_{\boldsymbol{\theta}} \sim \boldsymbol{\mathcal{G}}_{\boldsymbol{\theta}}} \left[\| \widehat{g}_{\boldsymbol{\theta}} \|^2 \right]$$

plus a constant[1] for unbiased estimators.

[1] $- \| \mathbb{E}_{\widehat{g}_{\boldsymbol{\theta}} \sim \boldsymbol{\mathcal{G}}_{\boldsymbol{\theta}}} [\widehat{g}_{\boldsymbol{\theta}}] \|^2$ to be precise

Instead of drawing a single sample $\widehat{g}_{\boldsymbol{\theta}}$, we can also draw N independent samples $\widehat{g}_{\boldsymbol{\theta}}^{(1)}, \ldots, \widehat{g}_{\boldsymbol{\theta}}^{(N)} \sim \boldsymbol{\mathcal{G}}_{\boldsymbol{\theta}}$ and take their average:

$$\frac{1}{N} \cdot \sum_{i=1}^{N} \widehat{g}_{\boldsymbol{\theta}}^{(i)}$$

N is sometimes known as the *batch size*.

This constitutes another estimator and an elementary calculation shows that the variance is decreased by a factor of $1/N$. However, this comes at the cost of more computations. As a consequence, to judge the quality of estimators we also need to take the computational efficiency into account.

Furthermore, rather than comparing the (raw) variances of estimators in experiments, we will compare ***work-normalised variances*** [Glynn and Whitt, 1992], which are the product of the variance and computational cost. In particular, increasing the batch size by a factor of k does *not* change the work-normalised variance since also the computational cost increases by a factor of k.

In summary, there are three[2] desiderata for gradient estimators: computational efficiency, unbiasedness and low variance.

Correctness of Stochastic Gradient Descent

Having discussed desirable properties of gradient estimators, we may wonder whether they are sufficient for the correctness of stochastic gradient descent and, in fact, what we actually mean by "correctness".

Since the objective function $\boldsymbol{\theta} \mapsto \mathbb{E}_{\boldsymbol{z} \sim \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}[f(\boldsymbol{\theta}, \boldsymbol{z})]$ may not be a convex function in the parameters $\boldsymbol{\theta}$, we cannot hope to always find global optima[3]. We seek ***stationary points*** instead, where the gradient w.r.t. the parameters $\boldsymbol{\theta}$ vanishes: $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z} \sim \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}[f(\boldsymbol{\theta}, \boldsymbol{z})] = \mathbf{0}$. In particular, this implicitly assumes that the objective function is differentiable.

Besides, recall that the iterants $\boldsymbol{\theta}_k$ are random variables dependent on $\widehat{g}_{\boldsymbol{\theta}_0}, \ldots, \widehat{g}_{\boldsymbol{\theta}_{k-1}}$, which means that correctness properties will be probabilistic. In the present thesis, we focus on

$$\liminf_{k \to \infty} \|\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}_k)\| = 0 \qquad \text{almost surely}$$

i.e. almost surely gradients will become arbitrarily small[4].

Next, we discuss further conditions beyond unbiasedness (Eq. (2.1)) to guarantee correctness in this sense. Most of the attention seems to be focussed on the step size sequence. The classical ***Robbins-Monro*** criterion introduced in the original paper [Robbins and Monro, 1951] states

$$\sum_{k \in \mathbb{N}} \gamma_k = \infty \qquad \sum_{k \in \mathbb{N}} \gamma_k^2 < \infty \tag{2.2}$$

[2] Sometimes *consistency* is added to this list. This property ensures convergence of an estimator to the true value as the batch size is increased. We will not discuss this notion in the remainder of the thesis.

[3] Indeed, the problem is NP-complete in general [Mahajan et al., 2012].

[4] A slightly stronger guarantee would be $\mathbb{P}\left[\lim_{k \to \infty} \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}_k) = \mathbf{0}\right] = 1$, i.e. almost surely *all* gradients will become arbitrarily small. This property is much harder to prove (especially for Diagonalisation Gradient Descent, see Chapter 6) and therefore we do not pursue it in the remainder of this thesis.

Intuitively, the first constraint ensures that we can make progress whilst the second ensures that the noise due to the estimation is balanced. This is satisfied by e.g. $\gamma_k \in \Theta(1/k)$.

The following result (the proof of which is standard) provides sufficient conditions for the convergence of SGD to stationary points (see e.g. [Bertsekas and Tsitsiklis, 2000] or [Bertsekas, 2015, Chapter 2]):

Proposition 2.11 (Correctness)**.** *Suppose* $\boldsymbol{\Theta} \subseteq \mathbb{R}^m$ *is convex,* $(\gamma_k)_{k\in\mathbb{N}}$ *satisfies Eq.* (2.2)*,*

$$g(\boldsymbol{\theta}) \coloneqq \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[f(\boldsymbol{\theta}, \boldsymbol{s})]$$

is well-defined and differentiable, and

(SGD1) $\nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}}[\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}, \boldsymbol{s})]$ *for all* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$
(SGD2) g *is bounded, Lipschitz continuous and Lipschitz smooth on* $\boldsymbol{\Theta}$
(SGD3) $\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[\|\nabla_{\boldsymbol{\theta}}\, f(\boldsymbol{\theta}, \boldsymbol{s})\|^2\right] < \infty$

Then almost surely $\liminf_{i\to\infty} \|\nabla g(\boldsymbol{\theta}_i)\| = 0$ *or* $\boldsymbol{\theta}_i \notin \boldsymbol{\Theta}$ *for some* $i \in \mathbb{N}$*.*

In Chapter 6 we prove a generalisation of this result (Proposition 6.1), the correctness of our *Diagonalisation* Stochastic Gradient Descent method.

Recall that a function $g : \boldsymbol{\Theta} \to \mathbb{R}$ is

– L-***Lipschitz continuous*** if for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \boldsymbol{\Theta}$,

$$|g(\boldsymbol{\theta}) - g(\boldsymbol{\theta}')| \leq L \cdot \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$$

– L-***Lipschitz smooth*** if it is differentiable and for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \boldsymbol{\Theta}$,

$$\|\nabla g(\boldsymbol{\theta}) - \nabla g(\boldsymbol{\theta}')\| \leq L \cdot \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$$

Furthermore, due to unbiasedness (premise (SGD1)), (SGD3) requires that the estimator

$$\nabla_{\boldsymbol{\theta}}\, f(\boldsymbol{\theta}, \boldsymbol{s}) \qquad\qquad \text{where } \boldsymbol{s} \sim \boldsymbol{\mathcal{D}}$$

has finite variance.

In practice, more complicated variants of the basic scheme of (SGD) are frequently employed, which choose the step size adaptively or use momentum, i.e. not only an estimate of the current gradient at $\boldsymbol{\theta}_k$ but also historic gradient information is used [Duchi et al., 2011, Hinton, 2012, Kingma and Ba, 2015].

Adam [Kingma and Ba, 2015] is one of the most successful optimisers in practice but is known to not always converge [Reddi et al., 2018]. Therefore, we will prove theoretical results of variants of (SGD) using assumptions resembling the classical choice of Eq. (2.2) but conduct experiments using Adam.

There are also ongoing efforts to relax some of the conditions in Proposition 2.11 in specific situations (e.g. [Moreau and Aeyels, 2000, Benaïm et al., 2005, Faw et al., 2022]), sometimes by complicating the basic iteration (SGD).

2.4.1 Score Estimator

When trying to derive an unbiased estimator of $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})}[f(\boldsymbol{\theta},\boldsymbol{z})]$ we need to express this quantity as an expectation $\mathbb{E}_{\widehat{g}_{\boldsymbol{\theta}}\sim\boldsymbol{\mathcal{G}}_{\boldsymbol{\theta}}}[\widehat{g}_{\boldsymbol{\theta}}]$ (not headed by a gradient evaluation). Exchanging differentiation and expectations/integrals, we can massage the gradient as follows:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})}[f(\boldsymbol{\theta},\boldsymbol{z})] &= \nabla_{\boldsymbol{\theta}} \int \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot f(\boldsymbol{\theta},\boldsymbol{z}) \,\mathrm{d}\boldsymbol{z} = \int \nabla_{\boldsymbol{\theta}} (\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot f(\boldsymbol{\theta},\boldsymbol{z})) \,\mathrm{d}\boldsymbol{z} \\ &= \int (\nabla_{\boldsymbol{\theta}} \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})) \cdot f(\boldsymbol{\theta},\boldsymbol{z}) + \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta},\boldsymbol{z}) \,\mathrm{d}\boldsymbol{z}\end{aligned}$$

We have made some progress since $\int \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta},\boldsymbol{z}) \,\mathrm{d}\boldsymbol{z} = \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})}[\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta},\boldsymbol{z})]$ can be estimated by the Monte Carlo method. On the other hand, it is not clear how to estimate $\int (\nabla_{\boldsymbol{\theta}} \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})) \cdot f(\boldsymbol{\theta},\boldsymbol{z}) \,\mathrm{d}\boldsymbol{z}$ because $\nabla_{\boldsymbol{\theta}} \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})$ is not necessarily a probability density (let alone one from which we can sample efficiently). Fortunately, we can employ the so-called *log-derivative trick*, which is the elementary insight that

$$\nabla_{\boldsymbol{\theta}} \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) = \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot \nabla_{\boldsymbol{\theta}} \log(\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}))$$

provided that $\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) \neq 0$ and arrive at

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})}[f(\boldsymbol{\theta},\boldsymbol{z})] &= \int \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot (\nabla_{\boldsymbol{\theta}} \log(\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}))) \cdot f(\boldsymbol{\theta},\boldsymbol{z}) \,\mathrm{d}\boldsymbol{z} + \int \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta},\boldsymbol{z}) \,\mathrm{d}\boldsymbol{z} \\ &= \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})} \left[(\nabla_{\boldsymbol{\theta}} \log(\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}))) \cdot f(\boldsymbol{\theta},\boldsymbol{z}) + \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta},\boldsymbol{z})\right]\end{aligned}$$

In summary, we can estimate $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})}[f(\boldsymbol{\theta},\boldsymbol{z})]$ via

$$\boxed{(\nabla_{\boldsymbol{\theta}} \log(\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}))) \cdot f(\boldsymbol{\theta},\boldsymbol{z}) + \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta},\boldsymbol{z}) \qquad \text{where } \boldsymbol{z} \sim \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) \quad \text{(Score)}}$$

This estimator is usually referred to as ***Score*** or ***REINFORCE***.

In particular, no (differentiability) assumption is necessary about f. However, despite popular belief the estimator may be biased if the density $\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}$ is not differentiable:

Example 2.12 ([Mohamed et al., 2020])**.** For $\mathcal{D}_\theta(z) \coloneqq \mathcal{U}_{(0,\theta)}(z) = \frac{1}{\theta}$ and $f_\theta(z) \coloneqq z$,

$$\mathbb{E}_{z\sim\mathcal{D}_\theta}\left[\underbrace{(\nabla_\theta \log(\mathcal{D}_\theta(z)))}_{=-1/\theta \text{ for } 0<z<\theta} \cdot f_\theta(z) + \underbrace{\nabla_\theta f_\theta(z)}_{=0}\right] = -\frac{1}{2} \neq \frac{1}{2} = \nabla_\theta \underbrace{\mathbb{E}_{z\sim\mathcal{D}_\theta}[f_\theta(z)]}_{=\frac{\theta}{2}}$$

2.4.2 Reparameterisation Estimator

Another approach—the ***reparameterisation gradient estimator***—reparameterises the latent variable $\boldsymbol{z}$ in terms of a base random variable $\boldsymbol{s}$ (viewed as the entropy source) via a diffeomorphic transformation $\boldsymbol{\varphi}_{\boldsymbol{\theta}}$. For example, if the distribution of the latent variable z is a Gaussian $\mathcal{N}(z \mid \mu, \sigma^2)$ with parameters $\boldsymbol{\theta} = \{\mu, \sigma\}$ then the location-scale transformation using the standard normal as the base distribution gives rise to the reparameterisation

$$z \sim \mathcal{N}(z \mid \mu, \sigma^2) \iff z = \varphi_{\mu,\sigma}(s), \quad s \sim \mathcal{N}(0,1). \tag{2.3}$$

where $\varphi_{\mu,\sigma}(s) \coloneqq s \cdot \sigma + \mu$. Besides, for the exponential distribution with rate parameter λ^{-1},

$$z \sim \mathcal{E}(z \mid \lambda^{-1}) \iff z \coloneqq s \cdot \lambda, \quad s \sim \mathcal{E}(1)$$

Moreover, in principle any distribution $\mathcal{D}$ on $\mathbb{R}$ can be reparameterised from the uniform distribution over $(0,1)$ via the inverse of its cumulative density function $f(z) = \int_{-\infty}^{z} \mathcal{D}(z')\,\mathrm{d}z'$:

$$z \sim \mathcal{D} \iff z = f^{-1}(s), \quad s \sim \mathcal{U}(0,1)$$

However, the (inverse of the) cumulative density function is usually not known in closed form. In particular, the Gamma and Beta distributions cannot be reparameterised efficiently, although there have been efforts to obtain reparameterisation-like estimators [Ruiz et al., 2016, Jankowiak and Obermeyer, 2018, Figurnov et al., 2018].

In general, we assume that

$$\boldsymbol{z} \sim \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) \iff \boldsymbol{z} = \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}), \quad \boldsymbol{s} \sim \boldsymbol{\mathcal{D}}$$

and therefore,

$$\mathbb{E}_{\boldsymbol{z} \sim \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}[f(\boldsymbol{\theta}, \boldsymbol{z})] = \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}}[f(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))] \tag{2.4}$$

The key advantage of this setup (often called "reparameterisation trick" [Kingma and Welling, 2014, Titsias and Lázaro-Gredilla, 2014, Rezende et al., 2014]) is that we have removed the dependency on $\boldsymbol{\theta}$ from the distribution w.r.t. which the expectation is taken. Therefore, we can now differentiate with respect to the parameters $\boldsymbol{\theta}$ of the variational distributions using a Monte Carlo estimation with draws $\boldsymbol{s}$ from the base distribution $\boldsymbol{\mathcal{D}}$. Thus, succinctly, we have

$$\nabla_{\boldsymbol{\theta}}\, \mathbb{E}_{\boldsymbol{z} \sim \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})}[f(\boldsymbol{\theta}, \boldsymbol{z})] = \nabla_{\boldsymbol{\theta}}\, \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}(\boldsymbol{s})}[f(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))] = \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}(\boldsymbol{s})}[\nabla_{\boldsymbol{\theta}}\, f(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))] \tag{2.5}$$

and we can estimate $\nabla_{\boldsymbol{\theta}}\, \mathbb{E}_{\boldsymbol{z} \sim \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})}[f(\boldsymbol{\theta}, \boldsymbol{z})]$ via

$$\nabla_{\boldsymbol{\theta}}\, f(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \qquad \text{where } \boldsymbol{s} \sim \boldsymbol{\mathcal{D}} \qquad \text{(Reparam)}$$

Variance

The reparameterisation gradient estimator is the dominant approach in practice for variational inference and other stochastic optimisation problems because it is folklore that empirically, it typically exhibits significantly lower variance than the Score estimator (e.g. [Mohamed et al., 2020, Rezende et al., 2014, Mohamed et al., 2020, Fu, 2006, Schulman et al., 2015, Xu et al., 2019, Lee et al., 2018]).

Example 2.13. Consider the setting where $f(z) \coloneqq 1$, $\varphi_\theta(s) = s + \theta$, $\mathcal{D}(s) = \mathcal{N}(s \mid 0, 1)$ and $\mathcal{D}_\theta(z) \coloneqq \mathcal{N}(z \mid \theta, 1)$. Then

$$\mathbb{E}_{s \sim \mathcal{N}(0,1)}\left[|\nabla_\theta f(\varphi_\theta(s))|^2\right] = \mathbb{E}_{s \sim \mathcal{N}(0,1)}[0] = 0$$

On the other hand

$$\mathbb{E}_{z \sim \mathcal{N}(\theta,1)}\left[|f(z) \cdot \nabla_\theta \log \mathcal{D}_\theta(z)|^2\right] = \mathbb{E}_{z \sim \mathcal{N}(\theta,1)}\left[(z-\theta)^2\right] = 1$$

However, the general reasons for this phenomenon are still poorly understood. One intuition is that for certain objective functions (which are independent sums for each dimension), the variance of Score grows linearly with the number of dimensions, whereas the variance of the reparameterisation estimator is constant [Rezende et al., 2014, Mohamed et al., 2020].

Example 2.14. Suppose $f(\boldsymbol{z}) = \sum_{j=1}^{n} f_j(z_j)$ and $\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}) = (\varphi_{\theta_1}^{(1)}(\boldsymbol{s}), \dots, \varphi_{\theta_n}^{(n)}(\boldsymbol{s}))$. Then

$$\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[\left|\frac{\partial}{\partial\theta_i} f(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))\right|^2\right] = \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[\left|\frac{\partial f_i}{\partial z_i}(\varphi_{\boldsymbol{\theta}}^{(i)}(\boldsymbol{s})) \cdot \frac{\partial \varphi_{(-)}^{(i)}}{\partial\theta_i}(\boldsymbol{\theta}, \boldsymbol{s})\right|^2\right]$$

and the variance is independent from the dimension n, whereas

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}\left[\left|\left(\frac{\partial \log \boldsymbol{\mathcal{D}}_{(-)}}{\partial\theta_i}(\boldsymbol{\theta}, \boldsymbol{z})\right) \cdot f(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s})) - \frac{\partial}{\partial\theta_i}\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[f(\boldsymbol{z})]\right|^2\right] \\
&= \sum_{j=1}^{n} \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}\left[\left|\left(\frac{\partial \log \boldsymbol{\mathcal{D}}_{(-)}}{\partial\theta_i}(\boldsymbol{\theta}, \boldsymbol{z})\right) \cdot f^{(j)}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s})) - \frac{\partial}{\partial\theta_i}\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[f_j(\boldsymbol{z})]\right|^2\right]
\end{aligned}$$

[Xu et al., 2019] give a theoretical proof for the superior variance of the reparameterisation gradient estimator in the special case of mean-field Gaussians and quadratic functions.

Despite the overwhelming empirical success of the reparameterisation estimator over the Score estimator, it is worth mentioning that examples do exist, where Score outperforms the reparameterisation estimator:

Example 2.15. Consider $f(z) \coloneqq \exp(-z^2)$, $\varphi_\theta(s) \coloneqq s + \theta$. Note that

$$\mathcal{N}(s \mid 0, 1) \cdot \exp\left(-2 \cdot (s+\theta)^2\right) = \mathcal{N}\left(s \mid -\frac{4}{5}\cdot\theta, \frac{1}{5}\right) \cdot \frac{1}{\sqrt{5}} \cdot \exp\left(-\frac{2}{5}\cdot\theta^2\right)$$

Thus, we obtain for the Score estimator:

$$\begin{aligned}
&\mathbb{E}_{z\sim\mathcal{N}(\theta,1)}\left[\left|(\nabla_{\boldsymbol{\theta}} \log \mathcal{N}(z \mid \theta, 1)) \cdot \exp\left(-z^2\right)\right|^2\right] \\
&= \mathbb{E}_{z\sim\mathcal{N}(\theta,1)}\left[(z-\theta)^2 \cdot \exp\left(-2\cdot z^2\right)\right] \\
&= \mathbb{E}_{s\sim\mathcal{N}(0,1)}\left[s^2 \cdot \exp\left(-2\cdot(s+\theta)^2\right)\right] \\
&= \frac{1}{\sqrt{5}} \cdot \exp\left(-\frac{2}{5}\cdot\theta^2\right) \cdot \mathbb{E}_{s\sim\mathcal{N}\left(-\frac{4}{5}\cdot\theta, \frac{1}{5}\right)}\left[s^2\right] \\
&= \frac{1}{\sqrt{5}} \cdot \exp\left(-\frac{2}{5}\cdot\theta^2\right) \cdot \left(\frac{16}{25}\cdot\theta^2 + \frac{1}{5}\right)
\end{aligned}$$

A similar calculation for the reparameterisation estimator yields:

$$\begin{aligned}
&\mathbb{E}_{s\sim\mathcal{N}(0,1)}\left[\left|\nabla_{\boldsymbol{\theta}} \exp\left(-(\varphi_\theta(s))^2\right)\right|^2\right] \\
&= \mathbb{E}_{s\sim\mathcal{N}(0,1)}\left[4\cdot(s+\theta)^2 \cdot \exp\left(-2\cdot(s+\theta)^2\right)\right] \\
&= \frac{1}{\sqrt{5}} \cdot \exp\left(-\frac{2}{5}\cdot\theta^2\right) \cdot 4 \cdot \mathbb{E}_{s\sim\mathcal{N}\left(-\frac{4}{5}\cdot\theta, \frac{1}{5}\right)}\left[(s+\theta)^2\right]
\end{aligned}$$

```
let z = sample 𝒩(0,1)
in observe 0.7 from 𝒩(if z < 0 then −2 else 5, 1)
```

(a) Model

```
z = sample 𝒩(θ, 1)
```

(b) Guide

Figure 2.4: Counter-example from [Lee et al., 2018, Proposition 2].

$$= \frac{1}{\sqrt{5}} \cdot \exp\left(-\frac{2}{5} \cdot \theta^2\right) \cdot 4 \cdot \left(\frac{16}{25} \cdot \theta^2 + \frac{1}{5} - \frac{8}{5} \cdot \theta^2 + \theta^2\right)$$
$$= \frac{1}{\sqrt{5}} \cdot \exp\left(-\frac{2}{5} \cdot \theta^2\right) \cdot \left(\frac{4}{25} \cdot \theta^2 + \frac{4}{5}\right)$$

Consequently, the reparameterisation estimator has a strictly larger variance than Score for $\theta \leq 1$.

Bias of the Reparameterisation Gradient Estimator

Unfortunately, it is well-known that the reparameterisation gradient estimator may be biased for discontinuous f:

Example 2.16. The counterexample in [Lee et al., 2018, Proposition 2], where the objective function is the ELBO for a non-differentiable model (Fig. 2.4), can be simplified to

$$f(\theta, s) = -0.5 \cdot \theta^2 + \begin{cases} 0 & \text{if } s + \theta < 0 \\ 1 & \text{otherwise} \end{cases}$$

Observe that (see Fig. 2.5a):

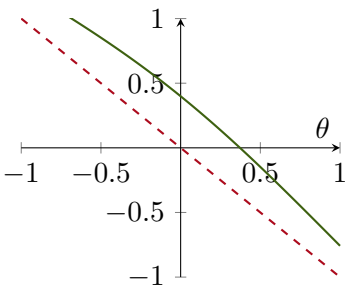
$$\nabla_\theta \, \mathbb{E}_{s \sim \mathcal{N}(0,1)} \left[f(\theta, s) \right] = -\theta + \mathcal{N}(-\theta \mid 0, 1) \neq -\theta = \mathbb{E}_{s \sim \mathcal{N}(0,1)} \left[\nabla_\theta f(\theta, s) \right]$$

The essence of the problem is that in the derivation of the estimator (Eq. (2.5)) we have blindly exchanged differentiation and expectations. However, this operation is only valid under certain circumstances. The following well-known result is a consequence of the dominated convergence theorem [Klenke, 2014, Theorem 6.28] and provides sufficient conditions:

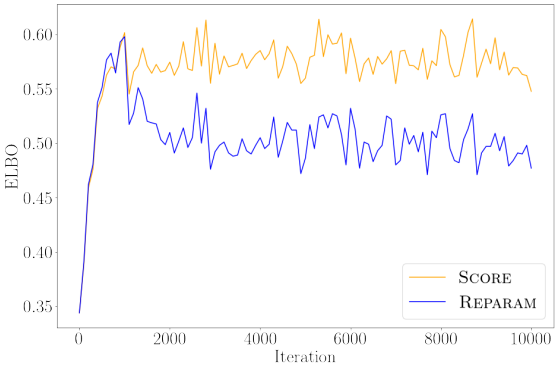
Lemma 2.17. *Let $\Theta \subseteq \mathbb{R}$ be open and $U \subseteq \mathbb{R}^n$ be measurable. If $f : \Theta \times U \to \mathbb{R}$ satisfies*

(i) *for each $\theta \in \Theta$, $\boldsymbol{s} \mapsto f(\theta, \boldsymbol{s})$ is integrable*
(ii) *$(\theta, \boldsymbol{s}) \mapsto f(\theta, \boldsymbol{s})$ is differentiable w.r.t. θ*
(iii) *there exists an integrable $g : U \to \mathbb{R}$ satisfying $\left|\frac{\partial f}{\partial \theta}(\theta, \boldsymbol{s})\right| \leq g(\boldsymbol{s})$ for all $(\theta, \boldsymbol{s}) \in \Theta \times U$*

then for all $\theta \in \Theta$, $\frac{\partial}{\partial \theta} \int_U f(\theta, \boldsymbol{s}) \, \mathrm{d}\boldsymbol{s} = \int_U \frac{\partial f}{\partial \theta}(\theta, \boldsymbol{s}) \, \mathrm{d}\boldsymbol{s}$.



(a) Dashed red: biased estimator $\mathbb{E}_{s\sim\mathcal{N}(0,1)}[\nabla_\theta f(\theta, s)]$, solid green: true gradient $\nabla_\theta \mathbb{E}_{s\sim\mathcal{N}(0,1)}[f(\theta, s)]$.



(b) ELBO trajectories (higher means better) obtained with our implementation (cf. Chapter 7)

Figure 2.5: Bias of the reparameterisation gradient estimator for Example 2.16.

Note that the second premise fails for the function f in Example 2.16: for *all* $s \in \mathbb{R}$, $\frac{\partial f}{\partial \theta}(-s, s)$ does not exist.

As a consequence of violating unbiasedness—one of the premises of Proposition 2.11—even if stochastic gradient descent converges, the gradient of the expectation may not vanish in the limit, potentially resulting in suboptimal solutions. (This phenomenon can indeed be observed when running our prototypical implementation of Chapter 7 on Example 2.16, see Fig. 2.5b.)

The objective of the Chapters 5 and 6 is to develop stochastic optimisation methods, which are computationally efficient, converge quickly and are provably correct. In particular, we wish to use estimators similar to the reparameterisation estimator for non-differentiable and discontinuous models.

2.5 Schwartz Densities and Polynomially-Bounded Functions

Before trying to solve the Optimisation Problem 2.10, we ought to take a step back and acknowledge that it is not *a priori* clear that the problem is well-defined. For instance, take $f(z) := \exp(z^2)$: we have $\mathbb{E}_{z\sim\mathcal{N}}[f(z)] = \infty$, independently of parameters. Therefore, we aim to guarantee:

$$\mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}[|f(\boldsymbol{\theta}, \boldsymbol{z})|] < \infty \qquad \text{for all } \boldsymbol{\theta} \in \boldsymbol{\Theta}$$

Issues may be caused not only by f but also by the distributions $\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}$. For example the Cauchy distribution does not even have an expectation.

Therefore, we discuss abstract conditions on the integrant f and distribution $\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}$, which guarantee a well-defined objective functions and which will guide the design of our programming languange (in Section 4.5).

We slightly generalise the well-behaved class of Schwartz functions (see also for more details e.g. [Hörmander, 2015, Reed and Simon, 2003]) to accommodate probability density functions the support of which is a subset of $\mathbb{R}^n$:

Definition 2.18. A function $f : U \to \mathbb{R}$, where $U \subseteq \mathbb{R}^n$ is measurable and $\text{Leb}(\partial U) = 0$, is a (generalised) ***Schwartz function*** if $f_{|\mathring{U}}$ is smooth and for all α and β (using standard multi-index notation for higher-order partial derivatives),

$$\sup_{\boldsymbol{x} \in \mathring{U}} \left| \boldsymbol{x}^\beta \cdot \partial^\alpha f(\boldsymbol{x}) \right| < \infty$$

Intuitively, a Schwartz function decreases rapidly.

In this work we are particularly interested in Schwartz functions which are also the probability density function of a continuous probability distribution.

Example 2.19. Distributions with pdfs which are also Schwartz functions include (for a fixed parameter) the (half) normal, exponential and logistic distributions (see Table 2.1). For the standard normal distribution, which has the pdf

$$f(x) = \frac{1}{\sqrt{2\pi}} \cdot \exp\left(-\frac{x^2}{2}\right)$$

it suffices to observe that derivatives of all orders have the form $f(x) \cdot p(x)$ for some polynomial $p : \mathbb{R} \to \mathbb{R}$.

Non-examples include the Cauchy distribution and the Gamma distributions. The Cauchy distribution famously does not even have an expectation (which would otherwise contradict Lemma 2.20 below). Even the pdf itself of the Gamma distribution for shape parameter $k < 1$ is unbounded. However, it cannot be reparameterised [Ruiz et al., 2016] and therefore it is only of marginal interest for our work regarding the reparameterisation gradient.

The following pleasing properties of Schwartz functions [Hörmander, 2015, Reed and Simon, 2003] carry immediately over:

Lemma 2.20. *Let $f : U \to \mathbb{R}$ be a Schwartz function.*

(i) *All partial derivatives of f are also Schwartz functions.*
(ii) *$f \in L^p(U)$; in particular f is integrable: $\int_U |f(\boldsymbol{x})| \, \mathrm{d}\boldsymbol{x} < \infty$.*
(iii) *The product $(f \cdot p) : U \to \mathbb{R}$ is also a Schwartz function if $p : \mathbb{R}^n \to \mathbb{R}$ is a polynomial.*

Next, we define the following straightforward notions:

Definition 2.21. (i) Let $f : U \to \mathbb{R}$, where $U \subseteq \mathbb{R}^n$. f is ***bounded by a polynomial*** if there exists a polynomial $p : \mathbb{R}^n \to \mathbb{R}$ satisfying $|f(\boldsymbol{x})| \leq p(\boldsymbol{x})$ for all $\boldsymbol{x} \in U$.

(ii) Let $f : \boldsymbol{\Theta} \times U \to \mathbb{R}$, where $\boldsymbol{\Theta} \subseteq \mathbb{R}^m$ and $U \subseteq \mathbb{R}^n$. f is ***uniformly bounded by a polynomial*** if there exists a polynomial $p : \mathbb{R}^n \to \mathbb{R}$ satisfying $|f(\boldsymbol{\theta}, \boldsymbol{x})| \leq p(\boldsymbol{x})$ for all $(\boldsymbol{\theta}, \boldsymbol{x}) \in \boldsymbol{\Theta} \times U$.

Example 2.22. Let $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}_{>0}$. The log-normal pdf

$$f(x) \coloneqq \log \mathcal{N}(x \mid \mu, \sigma^2) = -\log\left(\sigma \cdot \sqrt{2\pi}\right) - \frac{1}{2} \cdot \left(\frac{x - \mu}{\sigma}\right)^2$$

is bounded by the polynomial $p(x) \coloneqq c_1 + c_2 \cdot (x^2 + 1 + c_3)^2$, where

$$c_1 \coloneqq |\log(\sigma \cdot \sqrt{2\pi})| \qquad c_2 \coloneqq \frac{1}{2 \cdot \sigma^2} \qquad c_3 \coloneqq |\mu|$$

On the other hand, $g(x, \mu, \sigma) \coloneqq \log \mathcal{N}(x \mid \mu, \sigma^2)$ *cannot* be bounded by a polynomial because $\lim_{\sigma \searrow 0} g(x, \sigma, x) = \infty$.

The following result is an immediate consequence of Lemma 2.20 and it is useful for demonstrating that the objective functions of stochastic optimisation problems are well-defined:

Corollary 2.23 (Integrability)**.** *Let $\mathcal{D}$ be a distribution with a Schwartz density function and let $f : \Theta \times \mathbb{R}^n \to \mathbb{R}$ be bounded by a polynomial. Then for all $\boldsymbol{\theta} \in \Theta$,*

$$\mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}}[f(\boldsymbol{\theta}, \boldsymbol{s})] \leq \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}}[|f(\boldsymbol{\theta}, \boldsymbol{s})|] < \infty$$

Besides, the following can be viewed as an extension of Lemma 2.20(ii) (and it will be useful in Sections 5.3 and 6.2):

Lemma 2.24. *If $f : \Theta \times U \to \mathbb{R}$ is continuous and satisfies*

$$\sup_{(\boldsymbol{\theta},\boldsymbol{z})\in\Theta\times U} \|\boldsymbol{z}\|^{n+3} \cdot |f(\boldsymbol{\theta}, \boldsymbol{z})| < \infty$$

then $\int_U \sup_{\boldsymbol{\theta}\in\Theta} |f(\boldsymbol{\theta}, \boldsymbol{z})| \,\mathrm{d}\boldsymbol{z} < \infty$.

Proof. Let $g(\boldsymbol{z}) \coloneqq \sup_{\boldsymbol{\theta}\in\Theta} |f(\boldsymbol{\theta}, \boldsymbol{z})|$ and $U_k \coloneqq U \cap [-k, k]^n$. Then

$$\begin{aligned}
&\int_U g(\boldsymbol{z}) \,\mathrm{d}\boldsymbol{z} \\
&= \int_{U_1} g(\boldsymbol{z}) \,\mathrm{d}\boldsymbol{z} + \int_{U\setminus U_1} g(\boldsymbol{z}) \,\mathrm{d}\boldsymbol{z} \\
&\leq 2^n \cdot \sup_{(\boldsymbol{\theta},\boldsymbol{z})\in\Theta\times U_1} |f(\boldsymbol{\theta}, \boldsymbol{z})| + \left(\sup_{(\boldsymbol{\theta},\boldsymbol{z})\in\Theta\times U} \|\boldsymbol{z}\|^{n+3} \cdot f(\boldsymbol{\theta}, \boldsymbol{z}) \right) \cdot \int_{U\setminus U_1} \frac{1}{\|\boldsymbol{z}\|^{n+3}} \,\mathrm{d}\boldsymbol{z}
\end{aligned}$$

All the terms are finite because for $\boldsymbol{z} \in U \setminus U_k$, $k \leq \|\boldsymbol{z}\|$, and hence

$$\begin{aligned}
\int_{U\setminus U_1} \frac{1}{\|\boldsymbol{z}\|^{n+3}} \,\mathrm{d}\boldsymbol{z} &= \sum_{k=1}^{\infty} \int_{U_{k+1}\setminus U_k} \frac{1}{\|\boldsymbol{z}\|^{n+3}} \,\mathrm{d}\boldsymbol{z} \\
&\leq \sum_{k=1}^{\infty} \int_{U_{k+1}} \frac{1}{k^{n+3}} \,\mathrm{d}\boldsymbol{z} \\
&\leq \sum_{k=1}^{\infty} \frac{(2k)^{n+1}}{k^{n+3}} \,\mathrm{d}\boldsymbol{z} \\
&\leq 4^n \cdot \sum_{k=1}^{\infty} \frac{1}{k^2} \,\mathrm{d}\boldsymbol{z} < \infty
\end{aligned}$$

□

2.5.1 Polynomial Bounds

We list some useful technical properties of polynomials and functions bounded by polynomials (cf. Definition 2.21):

Lemma 2.25. (i) *If $f : \mathbb{R}^\ell \to \mathbb{R}$ and $f_1, \ldots, f_\ell : \mathbb{R}^n \to \mathbb{R}$ are bounded by polynomials then so is $f \circ \langle f_1, \ldots, f_\ell \rangle$.*

(ii) *If $f : \mathbb{R}^n \to \mathbb{R}$ (not necessarily bounded by polynomials) and $g, h : \mathbb{R}^n \to \mathbb{R}$ are bounded by polynomials then so is $[f(-) < 0] \cdot g + [f(-) \geq 0] \cdot h$.*

Here, $[\varphi]$ is the Iverson bracket (which evaluates to 1 if the condition φ holds and 0 otherwise).

The proof exploits the (first part of the) following elementary property:

Lemma 2.26. (i) *If $p : \mathbb{R}^n \to \mathbb{R}$ is a polynomial then there exists a polynomial q such that for all $|x_1| \leq |x'_1|, \ldots, |x_n| \leq |x'_n|$,*

$$|p(x_1, \ldots, x_n)| \leq q(x'_1, \ldots, x'_n)$$

(ii) *If $f : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$ is bounded by a polynomial, where $\boldsymbol{\Theta} \subseteq \mathbb{R}^m$ is bounded then it is* uniformly *bounded by a polynomial $p : \mathbb{R}^n \to \mathbb{R}$.*

For example for $p(x_1, x_2) = x_1^2 \cdot x_2 - x_2$ the polynomial $x_1^2 \cdot (x_2^2 + 1) + (x_2^2 + 1)$ satisfies this property. (The following proof yields $((x_1^2 + 1)^2 + 1) \cdot (x_2^2 + 1) + (x_2^2 + 1)$.) Besides, $p(\theta, x_1, x_2) = \theta \cdot x_1 \cdot x_2 - x_2$ is uniformly bounded by $2 \cdot (x_1^2 + 1) \cdot (x_2^2 + 1) + (x_2^2 + 1)$ on $\Theta = (-2, 1)$.

Proof. (i) If $p(\boldsymbol{x}) = x_i$ then we can choose $q(\boldsymbol{x}) \coloneqq (x_i^2 + 1)$ because for $|x_i| \leq |x'_i|$, $|x_i| \leq |x'_i| < (x'_i)^2 + 1$.
If $p(\boldsymbol{x}) = c$ for $c \in \mathbb{R}$ then we can choose $q(\boldsymbol{x}) \coloneqq |c|$.
Finally, suppose that p_1,p_2,q_1 and q_2 are polynomials such that for all $|x_1| \leq |x'_1|$, $\ldots, |x_n| \leq |x'_n|$, $|p_1(\boldsymbol{x})| \leq q_1(\boldsymbol{x}')$ and $|p_2(\boldsymbol{x})| \leq q_2(\boldsymbol{x}')$. Then for all $|x_1| \leq |x'_1|, \ldots,$ $|x_n| \leq |x'_n|$,

$$|p_1(\boldsymbol{x}) + p_2(\boldsymbol{x})| \leq |p_1(\boldsymbol{x})| + |p_2(\boldsymbol{x})| \leq q_1(\boldsymbol{x}') + q_2(\boldsymbol{x}')$$
$$|p_1(\boldsymbol{x}) \cdot p_2(\boldsymbol{x})| \leq |p_1(\boldsymbol{x})| \cdot |p_2(\boldsymbol{x})| \leq q_1(\boldsymbol{x}') \cdot q_2(\boldsymbol{x}')$$

(ii) If $f : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$ is bounded by a polynomial then by the first part there exists $q : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$ such that for all $(\boldsymbol{\theta}, \boldsymbol{z}), (\boldsymbol{\theta}', \boldsymbol{z}) \in \boldsymbol{\Theta} \times \mathbb{R}^n$ with $|\boldsymbol{\theta}| \leq |\boldsymbol{\theta}'|$, $|f(\boldsymbol{\theta}, \boldsymbol{z})| \leq q(\boldsymbol{\theta}, \boldsymbol{z}')$. Let $\theta_i^* \coloneqq \sup_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} |\theta_i| < \infty$ ($\boldsymbol{\Theta}$ is bounded) and $p(\boldsymbol{z}) \coloneqq q(\boldsymbol{z}, \boldsymbol{\theta}^*)$. Finally, it suffices to note that for every $(\boldsymbol{\theta}, \boldsymbol{z}) \in \boldsymbol{\Theta} \times \mathbb{R}^n$, $|f(\boldsymbol{\theta}, \boldsymbol{z})| \leq q(\boldsymbol{\theta}^*, \boldsymbol{z}) = p(\boldsymbol{z})$. □

Proof of Lemma 2.25. (i) By Lemma 2.26(i), there exists a polynomial $p : \mathbb{R}^\ell \to \mathbb{R}$ such that for all $|\boldsymbol{x}| \leq |\boldsymbol{x}'|$, $|f(\boldsymbol{x})| \leq p(\boldsymbol{x}')$. Furthermore, let $p_1, \ldots, p_\ell$ be polynomial bounds to the $f_1, \ldots, f_\ell$, i.e. $|f_i(\boldsymbol{x})| \leq p_i(\boldsymbol{x})$. Consequently, $p \circ \langle p_1, \ldots, p_\ell \rangle$ is a polynomial bound for $f \circ \langle f_1, \ldots, f_\ell \rangle$.

(ii) It suffices to note that $|[f(-) < 0] \cdot g + [f(-) \geq 0] \cdot h| \leq |g| + |h|$. □

In the same way as Lemma 2.25 we can prove:

Lemma 2.27. (i) *If $f : \mathbb{R}^\ell \to \mathbb{R}$ is bounded by a polynomial and $f_1, \ldots, f_\ell : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$ are* uniformly *bounded by polynomials then so is $f \circ \langle f_1, \ldots, f_\ell \rangle$.*

(ii) *If $f : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$ (not necessarily bounded by polynomials) and $g, h : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$ are* uniformly *bounded by polynomials then so is $[f(-) < 0] \cdot g + [f(-) \geq 0] \cdot h$.*

2.5.2 Results for Unbiasedness and Correctness of SGD

We conclude the section by providing useful results for proving unbiasedness and the correctness of SGD.

Proposition 2.28 (Unbiasedness)**.** *Suppose $\boldsymbol{\mathcal{D}} : U \to \mathbb{R}_{\geq 0}$ is a Schwartz density function, $\boldsymbol{\Theta} \subseteq \mathbb{R}^m$ is open, and $f : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$ is smooth such that f and its partial derivatives are uniformly bounded by a polynomial. Then for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$,*

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[f(\boldsymbol{\theta}, \boldsymbol{s})] = \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}, \boldsymbol{s})]$$

In particular, $\boldsymbol{\theta} \mapsto \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[f(\boldsymbol{\theta}, \boldsymbol{s})]$ is differentiable.

Proof. It suffices to consider the case $f : \Theta \times \mathbb{R}^n \to \mathbb{R}$ for open $\Theta \subseteq \mathbb{R}$. By assumption there exist polynomials $p_1, p_2 : \mathbb{R}^n \to \mathbb{R}$ such that for $(\theta, \boldsymbol{s}) \in \Theta \times \mathbb{R}^n$,

$$|f(\theta, \boldsymbol{s})| \leq p_1(\boldsymbol{s}) \qquad \left|\frac{\partial f}{\partial \theta}(\theta, \boldsymbol{s})\right| \leq p_2(\boldsymbol{s}) \tag{2.6}$$

We wish to apply Lemma 2.17 to $\boldsymbol{\mathcal{D}}(\boldsymbol{s}) \cdot f(\theta, \boldsymbol{s})$ and hence verify the premises:

(i) First, we note that by Lemma 2.20, $\int_U |\boldsymbol{\mathcal{D}}(\boldsymbol{s}) \cdot f(\theta, \boldsymbol{s})| \,\mathrm{d}\boldsymbol{s} \leq \int_U |\boldsymbol{\mathcal{D}}(\boldsymbol{s}) \cdot p_1(\boldsymbol{s})| \,\mathrm{d}\boldsymbol{s} < \infty$ for all $\theta \in \Theta$.
(ii) Besides, $\boldsymbol{\mathcal{D}}(\boldsymbol{s}) \cdot f(\theta, \boldsymbol{s})$ is differentiable on $\boldsymbol{\Theta} \times \mathring{U}$ by assumption.
(iii) Finally, the third premise follows from Eq. (2.6) because by Lemma 2.20 $\boldsymbol{\mathcal{D}}(\boldsymbol{s}) \cdot p_2(\boldsymbol{s})$ is integrable.

Consequently, Lemma 2.17 is applicable to $\boldsymbol{\mathcal{D}}(\boldsymbol{s}) \cdot f(\theta, \boldsymbol{s})$, and the claim follows. □

Proposition 2.29 (Applicability of SGD)**.** *Let $\boldsymbol{\mathcal{D}}$ be a Schwartz density function and $\boldsymbol{\Theta} \subseteq \mathbb{R}^m$ be open and convex. If $f : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$ is smooth such that its partial derivatives up to order 2 (incl. f itself) are uniformly bounded by polynomials then the function*

$$\begin{aligned} \boldsymbol{\Theta} &\to \mathbb{R} \\ \boldsymbol{\theta} &\mapsto \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[f(\boldsymbol{\theta}, \boldsymbol{s})] \end{aligned}$$

is well-defined, bounded, Lipschitz continuous and Lipschitz smooth. Furthermore, the variance

$$\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[\|\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}, \boldsymbol{s})\|^2\right]$$

is bounded. In particular, Proposition 2.11 is applicable.

Proof. Corollary 2.23 provides well-definedness.

By assumption there exist polynomials $p_1, p_2 : \mathbb{R}^n \to \mathbb{R}$ satisfying

$$\left|\frac{\partial f}{\partial \theta_i}(\boldsymbol{\theta}, \boldsymbol{s})\right| \leq p_1(\boldsymbol{s}) \qquad \left|\frac{\partial^2 f}{\partial \theta_i \partial \theta_j}(\boldsymbol{\theta}, \boldsymbol{s})\right| \leq p_2(\boldsymbol{s})$$

for all $(\boldsymbol{\theta}, \boldsymbol{s}) \in \boldsymbol{\Theta} \times \mathbb{R}^n$. To show that the variance is bounded we observe that

$$\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[\left|\frac{\partial f}{\partial \theta_i}(\boldsymbol{\theta}, \boldsymbol{s})\right|^2\right] \leq \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[\underbrace{(p_1(\boldsymbol{s}))^2}_{\text{polynomial}}\right] < \infty$$

For Lipschitz smoothness it is sufficient to bound partial derivatives of order 2. This can be demonstrated by exchanging integration and differentiation as in the proof of Proposition 2.28:

$$\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\left|\frac{\partial^2}{\partial\theta_i\partial\theta_j}\,\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[f(\boldsymbol{\theta},\boldsymbol{s})]\right| = \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[\left|\frac{\partial^2 f}{\partial\theta_i\partial\theta_j}(\boldsymbol{\theta},\boldsymbol{s})\right|\right] \leq \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[p_2(\boldsymbol{s})\right] < \infty$$

Boundedness and Lipschitz continuity can be verified similarly. □

2.6 Analytic Functions

Definition 2.30. A function $f : R^m \to R^n$ is ***analytic*** if it is infinitely differentiable and its multivariate Taylor expansion at every point $\boldsymbol{x}_0 \in \mathbb{R}^m$ converges pointwise to f in a neighbourhood of $\boldsymbol{x}_0$.

Recall the following dichotomy [Mityagin, 2015]:

Lemma 2.31. *If $f : \mathbb{R}^n \to \mathbb{R}$ is analytic then $f = 0$ (constantly) or $f \neq 0$ almost everywhere.*

Lemma 2.32. *Suppose $f : \mathbb{R}^\ell \rightharpoonup \mathbb{R}$ and $g_1 : \mathbb{R}^{n_1} \rightharpoonup \mathbb{R}, \ldots, g_\ell : \mathbb{R}^{n_\ell} \rightharpoonup \mathbb{R}$ are analytic such that $\nabla f \neq \mathbf{0}$ a.e. and $\nabla g_1, \ldots, \nabla g_\ell \neq 0$ a.e. Then*[5] *$\nabla(f \circ (g_1 \times \ldots \times g_\ell)) \neq 0$ a.e.*

The proof exploits the following auxiliary lemma:

Lemma 2.33. *If $f : \mathbb{R}^n \to \mathbb{R}$ is analytic and $\nabla f \neq 0$ a.e. then for all $\boldsymbol{x} \in \mathbb{R}^n$ and $\delta > 0$ there exists $\epsilon > 0$ satisfying $[f(x), f(x) + \epsilon) \subseteq f(\mathbf{B}_\delta(\boldsymbol{x}))$ or $(f(x) - \epsilon, f(x)] \subseteq f(\mathbf{B}_\delta(\boldsymbol{x}))$.*

($\mathbf{B}_r(\boldsymbol{x})$ is the ball of radius $r > 0$ centred at $\boldsymbol{x}$.)

Proof. Suppose $\boldsymbol{x} \in \mathbb{R}^n$ and $\delta > 0$. Since $\mathbf{B}_\delta(\boldsymbol{x})$ is not negligible there exists $\boldsymbol{x}' \in \mathbf{B}_\delta(\boldsymbol{x})$ such that $\nabla f(\boldsymbol{x}) \neq \mathbf{0}$. The result follows from continuous differentiability of f and the mean value theorem by elementary reasoning. □

Proof of Lemma 2.32. To simplify notation, we focus on the case $\ell = 2$, and abbreviate $g \coloneqq g_1 : \mathbb{R}^m \to \mathbb{R}$, $h \coloneqq g_2 : \mathbb{R}^n \to \mathbb{R}$.

By Lemma 2.33, $g(\mathbf{B}_1(\mathbf{0}))$ and $h(\mathbf{B}_1(\mathbf{0}))$ are not negligible. Therefore, there exist $x' = g(\boldsymbol{x})$ and $y' = h(\boldsymbol{y})$ such that $\nabla f(x', y') \neq \mathbf{0}$. We assume $\frac{\partial f}{\partial x'}(x', y') > 0$ (otherwise the reasoning is analogous). Since f is continuously differentiable, there exists $\epsilon > 0$ such that $\frac{\partial f}{\partial \boldsymbol{x}'}(x'', y'') > 0$ for all $x'' \in \mathbf{B}_\epsilon(x')$ and $y'' \in \mathbf{B}_\epsilon(y')$.

Therefore, by the chain rule and the assumption that $\nabla g \neq 0$ a.e. it follows $\frac{\partial g}{\partial x_i} \neq 0$ a.e. for some i. Consequently, $\frac{\partial(f\circ(g\times h))}{\partial x_i}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\partial f}{\partial x'}(g(\boldsymbol{x})) \cdot \frac{\partial g}{\partial x_i}(\boldsymbol{x})$ is not constant 0 and therefore (being analytic) different from 0 a.e. □

[5] $(f \circ (g_1 \times \ldots \times g_\ell))(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\ell) = f(g_1(\boldsymbol{x}_1), \ldots, g_\ell(\boldsymbol{x}_\ell))$

2.7 Convergence of Sequences of Functions

Let f_η ($\eta \in \mathbb{R}_{>0}$) be a sequence of functions $U \to \mathbb{R}$, where $\eta > 0$ and U is a set, and $f : U \to \mathbb{R}$. f_η ***converges uniformly*** to f as $\eta \searrow 0$ (notation: $f_\eta \xrightarrow{\text{unif.}} f$) if for all $\epsilon > 0$ there exists $\eta_0 > 0$ satisfying

$$|f_\eta(\boldsymbol{x}) - f(\boldsymbol{x})| < \epsilon \qquad \forall 0 < \eta < \eta_0, \boldsymbol{x} \in U$$

If (U, Σ_U) is measurable with measure $\mu : \Sigma_U \to [0, \infty]$ then f_η converges to f μ-***almost everywhere*** if there exists $V \in \Sigma_U$ satisfying $\mu(V) = 0$ and for all $\boldsymbol{x} \in U \setminus V$, $f_\eta(\boldsymbol{x}) \to f(\boldsymbol{x})$. We will focus on $U \subseteq \mathbb{R}^n$ together with the Lebesgue measure and use the notation $f_\eta \xrightarrow{\text{a.e.}} f$.

f_η converges to f μ-***almost uniformly*** (notation $f_\eta \xrightarrow{\text{a.u.}} f$) if for all $\epsilon > 0$ there exists $V \in \Sigma_U$ such that $\mu(V) < \epsilon$ and f_η converges uniformly to f on $U \setminus V$.

The following is a well-known consequence of Egorov's theorem [Wheeden et al., 1977]:

Lemma 2.34. *Let f_η be a sequence of functions $U \to \mathbb{R}^n$, where $\eta > 0$ and $U \subseteq \mathbb{R}^n$ is measurable, and $f : U \to \mathbb{R}$. If μ is a finite measure then $f_\eta \xrightarrow{\text{a.e.}} f$ iff $f_\eta \xrightarrow{\text{a.u.}} f$.*

Chapter 3

Almost Everywhere Differentiability

In this chapter we address the aforementioned research question selected by Hongseok Yang in his FSCD 2019 invited lecture [Yang, 2019] as one of three open problems in the field of semantics for probabilistic programs:

> *What class of statistical probabilistic programs have densities that are differentiable almost everywhere?*

This question is of practical interest, because many modern inference algorithms are "gradient-based": they exploit the derivative of the density function in order to optimise the approximation process. This includes the well-known methods of Hamiltonian Monte-Carlo [Duane et al., 1987, Neal, 2011] and stochastic variational inference [Hoffman et al., 2013, Ranganath et al., 2014, Blei et al., 2017, Kucukelbir et al., 2015]. However, these techniques can only be applied when the derivative exists "often enough", and thus, in the context of probabilistic programming, almost everywhere differentiability is often cited as a requirement for correctness [Zhou et al., 2019, Lee et al., 2020a].

The main result of this chapter (Theorem 3.18) states that the weight function and value function are *differentiable almost everywhere* (that is, everywhere but on a set of measure zero), provided the program is *almost surely terminating* in a suitable sense. Our result holds for the universal language of Section 2.2 with recursion and higher-order functions. This applies to our running example: the program in Fig. 1.1a (expressible in SPCF using primitive functions that satisfy Assumption 3.1 – see Example 2.2) is almost surely terminating. We emphasise that it follows immediately that *purely deterministic programs with real parameters* denote functions that are almost everywhere differentiable.

Points of non-differentiability exist largely because of *branching*, which typically arises in a program when the control flow reaches a conditional statement. Hence, our work is a study of the connections between the traces of a probabilistic program and its branching structure. To achieve this we introduce *stochastic symbolic execution*, a form of operational semantics for probabilistic programs, designed to identify sets of traces corresponding to the same control-flow branch. Roughly, a reduction sequence in this semantics corresponds to a control flow branch, and the rules additionally provide for every branch a symbolic expression of the trace density, parameterised by the outcome of the random draws that the branch contains. We obtain our main result in conjunction with a careful analysis of the branching structure of almost surely terminating programs.

This chapter is based on a collaboration, which resulted in the publication [Mak et al., 2021].

Distributions. To simplify the presentation, we henceforth assume that terms only sample from the uniform distribution on $(0,1)$, i.e. $\mathrm{Dist} = \{\mathcal{U}\}$. Thus, we abbreviate $\mathbf{sample}_{\mathcal{U}}$ to $\mathbf{sample}$. Furthermore, we can restrict the space of traces accordingly to $\mathbb{S}_n := (0,1)^n$ and $\mathbb{S} := \bigcup_{n\in\mathbb{N}} \mathbb{S}_n$. Sampling from other real-valued distributions can be obtained from $\mathcal{U}$ by applying the inverse of the distribution's cumulative distribution function. Alternatively, the proofs can be straightforwardly extended to admit more distributions directly.

3.1 Differentiability of the Weight and Value Functions

To reason about the differential properties of these functions we place ourselves in a setting in which differentiation makes sense. We start with some preliminaries.

3.1.1 Background on Differentiable Functions

Basic real analysis gives a standard notion of differentiability at a point $\boldsymbol{\theta} \in \mathbb{R}^n$ for functions between Euclidean spaces $\mathbb{R}^n \to \mathbb{R}^m$. In this context a function $f : \mathbb{R}^n \to \mathbb{R}^m$ is ***smooth*** on an open $U \subseteq \mathbb{R}^n$ if it has derivatives of all orders at every point of U. The theory of *differential geometry* (see e.g. the textbooks [Tu, 2011, Lee, 2013, Lee, 2009]) abstracts away from Euclidean spaces to *smooth manifolds*. We recall the formal definitions.

A topological space $\mathcal{M}$ is ***locally Euclidean at a point*** $x \in \mathcal{M}$ if x has a neighbourhood U such that there is a homeomorphism φ from U onto an open subset of $\mathbb{R}^n$, for some n. The pair $(U, \varphi : U \to \mathbb{R}^n)$ is called a ***chart*** (of dimension n). We say $\mathcal{M}$ is ***locally Euclidean*** if it is locally Euclidean at every point. A ***manifold*** $\mathcal{M}$ is a Hausdorff, second countable, locally Euclidean space.

Two charts, $(U, \varphi : U \to \mathbb{R}^n)$ and $(V, \psi : V \to \mathbb{R}^m)$, are ***compatible*** if the function $\psi \circ \varphi^{-1} : \varphi(U \cap V) \to \psi(U \cap V)$ is smooth, with a smooth inverse. An ***atlas*** on $\mathcal{M}$ is a family $\{(U_\alpha, \varphi_\alpha)\}$ of pairwise compatible charts that cover $\mathcal{M}$. A ***smooth manifold*** is a manifold equipped with an atlas.

It follows from the topological invariance of dimension that charts that cover a part of the same connected component have the same dimension. We emphasise that, although this might be considered slightly unusual, distinct connected components need not have the same dimension. This is important for our purposes: $\mathbb{S}$ is easily seen to be a smooth manifold since each connected component $\mathbb{S}_i$ is diffeomorphic to $\mathbb{R}^i$. It is also straightforward to endow the set Λ of SPCF terms with a (smooth) manifold structure. Following [Borgström et al., 2016] we view Λ as $\bigcup_{m\in\mathbb{N}} (\mathrm{SK}_m \times \mathbb{R}^m)$, where SK_m is the set of SPCF terms with no real constants and exactly m place-holders (a.k.a. *skeleton terms*) for numerals. Thus identified, we give Λ the countable disjoint union topology of the product topology of the discrete topology on SK_m and the standard topology on $\mathbb{R}^m$. Note that the connected components of Λ have the form $\{M\} \times \mathbb{R}^m$, with M ranging over SK_m, and m over $\mathbb{N}$. So in particular, the subspace $\Lambda_v \subseteq \Lambda$ of values inherits the manifold structure. We fix the Borel algebra of this topology to be the σ-algebra on Λ.

Given manifolds $(\mathcal{M}, \{U_\alpha, \varphi_\alpha\})$ and $(\mathcal{M}', \{V_\beta, \psi_\beta\})$, a function $f : \mathcal{M} \to \mathcal{M}'$ is ***differentiable*** at a point $x \in \mathcal{M}$ if there are charts $(U_\alpha, \varphi_\alpha)$ about x and (V_β, ψ_β) about $f(x)$ such that the composite $\psi_\beta \circ f \circ \varphi_\alpha^{-1}$ restricted to the open subset $\varphi_\alpha(f^{-1}(V_\beta) \cap U_\alpha)$ is differentiable at $\varphi_\alpha(x)$.

The definitions above are useful because they allow for a uniform presentation. But it is helpful to unpack the definition of differentiability in a few instances, and we see that they boil down to the standard sense in real analysis. Take an SPCF term M with free variables amongst $\theta_1, \dots, \theta_m$ (all of type $\mathbb{R}$), and $(\boldsymbol{r}, \boldsymbol{s}) \in \mathbb{R}^m \times \mathbb{S}_n$.

- The function $\mathrm{weight}_M : \mathbb{R}^m \times \mathbb{S} \to \mathbb{R}_{\geq 0}$ is differentiable at $(\boldsymbol{r}, \boldsymbol{s}) \in \mathbb{R}^m \times \mathbb{S}_n$ just if its restriction $\mathrm{weight}_M|_{\mathbb{R}^m \times \mathbb{S}_n} : \mathbb{R}^m \times \mathbb{S}_n \to \mathbb{R}_{\geq 0}$ is differentiable at $(\boldsymbol{r}, \boldsymbol{s})$.

- In case M is of type R, $\mathrm{value}_M : \mathbb{R}^m \times \mathbb{S} \to \Lambda_v^0 \cup \{\bot\}$ is in essence a partial function $\mathbb{R}^m \times \mathbb{S} \rightharpoonup \mathbb{R}$. Precisely value_M is differentiable at $(\boldsymbol{r}, \boldsymbol{s})$ just if for some open neighbourhood $U \subseteq \mathbb{R}^m \times \mathbb{S}_n$ of $(\boldsymbol{r}, \boldsymbol{s})$:

  (i) $\mathrm{value}_M(\boldsymbol{r'}, \boldsymbol{s'}) = \bot$ for all $(\boldsymbol{r'}, \boldsymbol{s'}) \in U$; or
  (ii) $\mathrm{value}_M(\boldsymbol{r'}, \boldsymbol{s'}) \neq \bot$ for all $(\boldsymbol{r'}, \boldsymbol{s'}) \in U$, and $\mathrm{value}'_M : U \to \mathbb{R}$ is differentiable at $(\boldsymbol{r}, \boldsymbol{s})$, where we define $\mathrm{value}'_M(\boldsymbol{r'}, \boldsymbol{s'}) := r''$ whenever $\mathrm{value}_M(\boldsymbol{r'}, \boldsymbol{s'}) = \underline{r''}$.

3.1.2 Why Almost Everywhere Differentiability Can Fail

Conditional statements break differentiability. This is easy to see with an example: the weight function of the term

$$\mathbf{score}(\mathbf{if}\,(\mathbf{sample} - \mathbf{sample}) < 0\,\mathbf{then}\,\underline{1}\,\mathbf{else}\,\underline{0})$$

is exactly the characteristic function of $\{[s_1, s_2] \in \mathbb{S} \mid s_1 - s_2 < 0\}$, which is not differentiable on the diagonal $\{[s, s] \in \mathbb{S}_2 \mid s \in (0, 1)\}$.

This function is however differentiable *almost everywhere*: the diagonal is an uncountable set but has Leb_2 measure zero in the space $\mathbb{S}_2$. Unfortunately, this is not true in general. Without sufficient restrictions, conditional statements also break almost everywhere differentiability. This can happen for two reasons.

Problem 1: Pathological Primitive Functions.

Recall that our definition of SPCF is parameterised by a set Op of primitive functions. It is tempting in this context to take Op to be the set of all differentiable functions, but this is too general, as we show now. Consider that for every $f : \mathbb{R} \to \mathbb{R}$ the term

$$\mathbf{score}(\mathbf{if}\,\underline{f}(\mathbf{sample}) < 0\,\mathbf{then}\,\underline{1}\,\mathbf{else}\,\underline{0})$$

has weight function the characteristic function of $\{[s_1] \in \mathbb{S} \mid f(s_1) < 0\}$. This function is non-differentiable at every $s_1 \in \mathbb{S}_1 \cap \partial f^{-1}(-\infty, 0)$: in every neighbourhood of s_1 there are s_1' and s_1'' such that $f(s_1') < 0$ and $f(s_1'') \geq 0$. One can construct a differentiable f for which this is *not* a measure zero set. (For example, there exists a non-negative function f which is zero exactly on a *fat* Cantor set, i.e., a Cantor-like set with strictly positive measure. See [Rudin, 1976, Ex. 5.21].)

Problem 2: Non-Terminating Runs.

Our language has recursion, so we can construct a term which samples a random number, halts if this number is in $\mathbb{Q} \cap [0, 1]$, and diverges otherwise. In pseudocode:

```
let rec enumQ p q r =
  if (r = p/q) then (score 1) else
    if (r < p/q) then
      enumQ p (q+1) r
    else
      enumQ (p+1) q r
in enumQ 0 1 sample
```

The induced weight function is the characteristic function of $\{[s_1] \in \mathbb{S} \mid s_1 \in \mathbb{Q}\}$; the set of points at which this function is non-differentiable is $\mathbb{S}_1$, which has measure 1.

We proceed to overcome Problem 1 by making appropriate assumptions on the set of primitives. We will then address Problem 2 by focusing on *almost surely terminating* programs.

3.1.3 Admissible Primitive Functions

One contribution of this work is to identify sufficient conditions for Op. We will show in Section 3.3 that our main result holds provided:

Assumption 3.1 (Admissible Primitive Functions)**.** Op is a set of partial, measurable functions $\mathbb{R}^\ell \rightharpoonup \mathbb{R}$ including all constant and projection functions which satisfies

(i) if $f : \mathbb{R}^\ell \rightharpoonup \mathbb{R}$ and $g_i : \mathbb{R}^m \rightharpoonup \mathbb{R}$ are elements of Op for $i = 1, \dots, \ell$, then $f \circ \langle g_i \rangle_{i=1}^{\ell} : \mathbb{R}^m \rightharpoonup \mathbb{R}$ is in Op
(ii) if $(f : \mathbb{R}^\ell \rightharpoonup \mathbb{R}) \in \mathrm{Op}$, then f is differentiable in the interior of $\operatorname{dom} f$
(iii) if $(f : \mathbb{R}^\ell \rightharpoonup \mathbb{R}) \in \mathrm{Op}$, then $\mathrm{Leb}_\ell(\partial f^{-1}[0, \infty)) = 0$ and $\mathrm{Leb}_\ell(\partial \operatorname{dom} f) = 0$.

Example 3.2. The following sets of primitive operations satisfy the above sufficient conditions. (See Appendix A.1 for a proof.)

(i) The set Op_1 of analytic functions with co-domain $\mathbb{R}$ (see Section 2.6).
(ii) The set Op_2 of (partial) functions $f : \mathbb{R}^\ell \rightharpoonup \mathbb{R}$ such that $\operatorname{dom} f$ is open[1], and f is differentiable everywhere in $\operatorname{dom} f$, and $f^{-1}(I)$ is a finite union of (possibly unbounded) rectangles[2] for (possibly unbounded) intervals I.

Note that all primitive functions mentioned in our examples (and in particular the density of the normal distribution) are included in both Op_1 and Op_2.

It is worth noting that both Op_1 and Op_2 satisfy the following stronger (than Assumption 3.1(iii)) property: $\mathrm{Leb}_n(\partial f^{-1} I) = 0$ for every interval I, for every primitive function f.

3.1.4 Almost Sure Termination

To rule out the contrived counterexamples which diverge, we restrict attention to *almost surely terminating* SPCF terms. Intuitively, a program M (closed term of ground type) is almost surely terminating if the probability that a run of M terminates is 1.

Take an SPCF term M with variables amongst $\theta_1, \dots, \theta_m$ (all of type $\mathbb{R}$), and set

$$\mathbb{T}_{M,\mathrm{term}} \coloneqq \left\{ (\boldsymbol{r}, \boldsymbol{s}) \in \mathbb{R}^m \times \mathbb{S} \mid \exists V, w \,.\, \langle M[\boldsymbol{r}/\boldsymbol{\theta}], 1, [] \rangle \to^* \langle V, w, \boldsymbol{s} \rangle \right\}. \tag{3.1}$$

[1] This requirement is crucial, and cannot be relaxed.
[2] i.e. a finite union of $I_1 \times \cdots \times I_\ell$ for (possibly unbounded) intervals I_i

Let us first consider the case of closed $M \in \Lambda^0$ i.e. $m = 0$ (notice that the measure $\mu_{\mathbb{R}^m \times \mathbb{S}}$ is not finite, for $m \geq 1$). As $\mathbb{T}_{M,\mathrm{term}}$ now coincides with $\mathrm{value}_M^{-1}(\Lambda_v^0)$, $\mathbb{T}_{M,\mathrm{term}}$ is a measurable subset of $\mathbb{S}$. Plainly if M is deterministic (i.e. **sample**-free), then $\mu_{\mathbb{S}}(\mathbb{T}_{M,\mathrm{term}}) = 1$ if M converges to a value, and 0 otherwise. Generally for an arbitrary (stochastic) term M we can regard $\mu_{\mathbb{S}}(\mathbb{T}_{M,\mathrm{term}})$ as the probability that a run of M converges to a value, because of Lemma 3.3.

Lemma 3.3. *If $M \in \Lambda^0$ then $\mu_{\mathbb{S}}(\mathbb{T}_{M,\mathrm{term}}) \leq 1$.*

More generally, if M has free variables amongst $\theta_1, \ldots, \theta_m$ (all of type R), then we say that M is almost surely terminating if for almost every (instantiation of the free variables by) $\boldsymbol{r} \in \mathbb{R}^m$, $M[\boldsymbol{r}/\boldsymbol{\theta}]$ terminates with probability 1.

We formalise the notion of almost sure termination as follows.

Definition 3.4. Let M be an SPCF term. M ***terminates almost surely*** if

(i) M is closed and $\mu(\mathbb{T}_{M,\mathrm{term}}) = \mu(\mathrm{value}_M^{-1}(\Lambda_v^0)) = 1$; or
(ii) M has free variables amongst $\theta_1, \ldots, \theta_m$ (all of which are of type R), and there exists $T \in \Sigma_{\mathbb{R}^m}$ such that $\mathrm{Leb}_m(\mathbb{R}^m \setminus T) = 0$ and for each $\boldsymbol{r} \in T$, $M[\underline{\boldsymbol{r}}/\boldsymbol{\theta}]$ terminates almost surely.

Suppose that M is a closed term and $M^\flat$ is obtained from M by recursively replacing subterms $\mathbf{score}(L)$ with the term $(\lambda x.\, \mathbf{if}\, x < 0\, \mathbf{then}\, N_{\mathrm{fail}}\, \mathbf{else}\, x)\, L$, where x is a fresh variable and N_{fail} is a term that reduces to fail such as $\underline{1}/\underline{0}$. It is easy to see that for all $\boldsymbol{s} \in \mathbb{S}$, $\langle M^\flat, 1, [] \rangle \to^* \langle V, 1, \boldsymbol{s} \rangle$ iff for some (unique) $w \in \mathbb{R}_{\geq 0}$, $\langle M, 1, [] \rangle \to^* \langle V, w, \boldsymbol{s} \rangle$. Therefore,

$$\begin{aligned}\mu_{M^\flat}(\Lambda_v) &= \int_{\mathrm{value}_{M^\flat}^{-1}(\Lambda_v)} \mathrm{d}\mu_{\mathbb{S}}\ \mathrm{weight}_{M^\flat} \\ &= \mu_{\mathbb{S}}(\{\boldsymbol{s} \in \mathbb{S} \mid \exists V\,.\, \left\langle M^\flat, 1, [] \right\rangle \to^* \langle V, 1, \boldsymbol{s} \rangle\}) = \mu_{\mathbb{S}}(\mathbb{T}_{M,\mathrm{term}})\end{aligned}$$

Consequently, the closed term M terminates almost surely iff $\mu_{M^\flat}$ is a probability measure.

Remark 3.5.

– Like many treatments of semantics of probabilistic programs in the literature, we make no distinction between non-terminating runs and aborted runs of a (closed) term M: both could result in the value semantics $\mu_{M^\flat}$ being a sub-probabilty measure (cf. [Bichsel et al., 2018]).

– Even so, current probabilistic programming systems do not place any restrictions on the code that users can write: it is perfectly possible to construct invalid models because catching programs that do not define valid probability distributions can be hard, or even impossible. This is not surprising, because almost sure termination is hard to decide: it is Π_2^0-complete in the arithmetic hierarchy [Kaminski et al., 2019]. Nevertheless, because almost sure termination is an important correctness property of probabilistic programs (not least because of the main result of this chapter, Theorem 3.18), the development of methods to prove almost sure termination is a hot research topic.

Accordingly, the main theorem of this chapter is stated as follows:

Theorem 3.18. *Let M be an SPCF term (possibly with free variables of type R) which terminates almost surely. Then its weight function weight_M and value function value_M are differentiable almost everywhere.*

3.2 Stochastic Symbolic Execution

We have seen that a source of discontinuity is the use of if-statements. This result therefore relies on an in-depth understanding of the branching behaviour of programs. The operational semantics given in Section 2.2.2 is unsatisfactory in this respect: any two execution paths are treated independently, whether they go through different branches of an if-statement or one is obtained from the other by using slightly perturbed random samples not affecting the control flow.

More concretely, note that although we have derived $\mathrm{weight}_{\mathrm{Ped}}[0.2, 0.9, 0.7] = 0.54$ and $\mathrm{value}_{\mathrm{Ped}}[0.2, 0.9, 0.7] = \underline{0.6}$ in Example 2.3, we cannot infer anything about the slightly perturbed $\mathrm{weight}_{\mathrm{Ped}}[0.21, 0.91, 0.71]$ and $\mathrm{value}_{\mathrm{Ped}}[0.21, 0.91, 0.71]$ unless we perform the corresponding reduction.

So we propose an alternative *symbolic* operational semantics (similar to the "compilation scheme" in [Zhou et al., 2019]), in which no sampling is performed: whenever a **sample** command is encountered, we simply substitute a fresh variable α_i for it, and continue on with the execution. We can view this style of semantics as a stochastic form of symbolic execution [Clarke, 1976, King, 1976], i.e., a means of analysing a program so as to determine what *inputs*, and *random draws* (from **sample**) cause each part of a program to execute.

Consider the term $M \equiv \mathbf{let}\ x = \mathbf{sample} \cdot \underline{3}\ \mathbf{in}\ (\mathrm{walk}\, x)$, defined using the function walk of Example 2.2. We have a reduction path

$$M \Rightarrow \mathbf{let}\ (x = \alpha_1 \cdot \underline{3})\ \mathbf{in}\ (\mathrm{walk}\, x) \Rightarrow \mathrm{walk}\,(\alpha_1 \cdot \underline{3})$$

but at this point we are stuck: the CBV strategy requires a value for α_1. We will "delay" the evaluation of the multiplication $\alpha_1 \cdot \underline{3}$; we signal this by drawing a box around the delayed operation: $\alpha_1 \boxdot \underline{3}$. We continue the execution, inspecting the definition of walk, and get:

$$M \Rightarrow^* \mathrm{walk}\,(\alpha_1 \boxdot \underline{3}) \Rightarrow^* N \equiv \mathbf{if}\ \alpha_1 \boxdot \underline{3} \leq 0\ \mathbf{then}\ \underline{0}\ \mathbf{else}\ P$$

where

$$P \equiv \begin{pmatrix} \mathbf{let}\ s = \mathbf{sample}\ \mathbf{in} \\ \mathbf{if}\ \mathbf{sample} < 0.5\ \mathbf{then}\ \big(s + \mathrm{walk}(\alpha_1 \boxdot \underline{3} + s)\big)\ \mathbf{else}\ \big(s + \mathrm{walk}(\alpha_1 \boxdot \underline{3} - s)\big) \end{pmatrix}.$$

We are stuck again: the value of α_1 is needed in order to know which branch to follow. Our approach consists in considering the space $\mathbb{S}_1 = (0, 1)$ of possible values for α_1, and splitting it into $\{s_1 \in (0,1) \mid s_1 \cdot 3 \leq 0\} = \emptyset$ and $\{s_1 \in (0,1) \mid s_1 \cdot 3 > 0\} = (0, 1)$. Each of the two branches will then yield a weight function restricted to the appropriate subspace.

Formally, our symbolic operational semantics is a rewrite system of configurations of the form $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle$, where $\mathcal{M}$ is a term with delayed (boxed) operations, and free "sampling" variables[3] $\alpha_1, \ldots, \alpha_n$; $U \subseteq \mathbb{S}_n$ is the subspace of sampling values compatible with the current branch; and $w : U \to \mathbb{R}_{\geq 0}$ is a function assigning to each $\boldsymbol{s} \in U$ a weight $w(\boldsymbol{s})$. In particular, for our running example[4]

$$\langle\!\langle M, \mathbb{\lambda}[].\, 1, \mathbb{S}_0 \rangle\!\rangle \Rightarrow^* \langle\!\langle N, \mathbb{\lambda}[s_1].\, 1, (0,1) \rangle\!\rangle.$$

[3] Note that $\mathcal{M}$ may be open and contain other free "non-sampling" variables, usually denoted $\theta_1, \ldots, \theta_m$.

[4] We use the meta-lambda-abstraction $\mathbb{\lambda}x.\, f(x)$ to denote the set-theoretic function $x \mapsto f(x)$.

As explained above, this leads to two branches:

$$\langle\!\langle N, \mathbb{\lambda}[s_1].\,1, (0,1)\rangle\!\rangle \overset{*}{\Rightarrow} \langle\!\langle \underline{0}, \mathbb{\lambda}[s_1].\,1, \emptyset\rangle\!\rangle \\ \overset{*}{\Rightarrow} \langle\!\langle P, \mathbb{\lambda}[s_1].\,1, (0,1)\rangle\!\rangle$$

The first branch has reached a value, and the reader can check that the second branch continues as

$$\langle\!\langle P, \mathbb{\lambda}[s_1].\,1, (0,1)\rangle\!\rangle \Rightarrow^* \\ \langle\!\langle \textbf{if } \alpha_3 < 0.5 \textbf{ then } \alpha_2 + \mathrm{walk}(\alpha_1 \boxdot \underline{3} + \alpha_2) \textbf{ else } \alpha_2 + \mathrm{walk}(\alpha_1 \boxdot \underline{3} - \alpha_2), \mathbb{\lambda}[s_1, s_2, s_3].\,1, (0,1)^3\rangle\!\rangle$$

where α_2 and α_3 stand for the two **sample** statements in P. From here we proceed by splitting $(0,1)^3$ into $(0,1) \times (0,1) \times (0, 0.5]$ and $(0,1) \times (0,1) \times (0.5, 1)$ and after having branched again (on whether we have passed 0) the evaluation of walk can terminate in the configuration

$$\langle\!\langle \alpha_2 \boxed{+} 0, \mathbb{\lambda}[s_1, s_2, s_3].\,1, U\rangle\!\rangle$$

where $U \coloneqq \{[s_1, s_2, s_3] \in \mathbb{S}_3 \mid s_3 \geq 0.5 \wedge s_1 \cdot 3 - s_2 \leq 0\}$.

Recall that M appears in the context of our running example Ped. Using our calculations above we derive one of its branches:

$$\begin{aligned} &\langle\!\langle \mathrm{Ped}, \mathbb{\lambda}[].\,1, \{[]\}\rangle\!\rangle \Rightarrow^* \langle\!\langle \textbf{let } w = \textbf{score}(\underline{\mathrm{pdf}_{\mathcal{N}(1.1,0.1)}}(\alpha_2)) \textbf{ in } \alpha_1 \boxdot \underline{3}, \mathbb{\lambda}[s_1, s_2, s_3].\,1, U\rangle\!\rangle \\ &\quad \Rightarrow \langle\!\langle \textbf{let } w = \textbf{score}(\boxed{\mathrm{pdf}_{\mathcal{N}(1.1,0.1)}}(\alpha_2)) \textbf{ in } \alpha_1 \boxdot \underline{3}, \mathbb{\lambda}[s_1, s_2, s_3].\,1, U\rangle\!\rangle \\ &\quad \Rightarrow^* \langle\!\langle \textbf{let } w = \boxed{\mathrm{pdf}_{\mathcal{N}(1.1,0.1)}}(\alpha_2) \textbf{ in } \alpha_1 \boxdot \underline{3}, \mathbb{\lambda}[s_1, s_2, s_3].\,\mathrm{pdf}_{\mathcal{N}(1.1,0.1)}(s_2), U\rangle\!\rangle \\ &\quad \Rightarrow^* \langle\!\langle \alpha_1 \boxdot \underline{3}, \mathbb{\lambda}[s_1, s_2, s_3].\,\mathrm{pdf}_{\mathcal{N}(1.1,0.1)}(s_2), U\rangle\!\rangle \end{aligned}$$

In particular the trace $[0.2, 0.9, 0.7]$ of Example 2.3 lies in the subspace U. We can immediately read off the corresponding value and weight functions for *all* $[s_1, s_2, s_3] \in U$ simply by evaluating the computation $\alpha_1 \cdot \underline{3}$, which we have delayed until now:

$$\mathrm{value}_{\mathrm{Ped}}[s_1, s_2, s_3] = \underline{s_1 \cdot 3} \qquad \mathrm{weight}_{\mathrm{Ped}}[s_1, s_2, s_3] = \mathrm{pdf}_{\mathcal{N}(1.1,0.1)}(s_2)$$

3.2.1 Symbolic Terms and Values

We have just described informally our symbolic execution approach, which involves delaying the evaluation of primitive operations. We make this formal by introducing an extended notion of terms, which we call ***symbolic terms*** and define in Fig. 3.1a along with a notion of ***symbolic values***. For this we assume fixed denumerable sequences of *distinguished* variables: $\alpha_1, \alpha_2, \ldots$, used to represent sampling, and $\theta_1, \theta_2, \ldots$ used for free variables of type R. Symbolic terms are typically denoted $\mathcal{M}$, $\mathcal{N}$, or $\mathcal{L}$. They contain terms of the form $\boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell)$ for primitive functions $f : \mathbb{R}^\ell \rightharpoonup \mathbb{R} \in \mathrm{Op}$, representing delayed evaluations, and they also contain the sampling variables α_j. The type system is adapted in a straightforward way, see Fig. 3.1b.

We use $\Lambda_{(m,n)}$ to refer to the set of well-typed symbolic terms with free variables amongst $\theta_1, \ldots, \theta_m$ and $\alpha_1, \ldots, \alpha_n$ (and all are of type R). Note that every term in the sense of Section 2.2.1 is also a symbolic term.

$$\mathcal{V} ::= \underline{r} \mid \theta_i \mid \alpha_j \mid \boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell) \mid \lambda y.\, \mathcal{M}$$
$$\mathcal{M}, \mathcal{N}, \mathcal{L} ::= \mathcal{V} \mid y \mid \underline{f}(\mathcal{M}_1, \ldots, \mathcal{M}_\ell) \mid \mathbf{if}\ \mathcal{L} < 0\, \mathbf{then}\, \mathcal{M}\, \mathbf{else}\, \mathcal{N}$$
$$\mid \mathcal{M}\, \mathcal{N} \mid \mathbf{Y}\, \mathcal{M}$$
$$\mid \mathbf{sample} \mid \mathbf{score}(\mathcal{M})$$

(a) Symbolic values (typically $\mathcal{V}$) and symbolic terms (typically $\mathcal{M}$, $\mathcal{N}$ or $\mathcal{L}$)

$$\frac{\Gamma \vdash \mathcal{V}_1 : R \quad \cdots \quad \Gamma \vdash \mathcal{V}_\ell : R}{\Gamma \vdash \boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell) : R} \qquad \frac{}{\Gamma \vdash \theta_i : R} \qquad \frac{}{\Gamma \vdash \alpha_j : R}$$

$$\frac{}{\Gamma, y : \tau \vdash y : \tau} \qquad \frac{}{\Gamma \vdash \underline{r} : R}\ r \in \mathbb{R} \qquad \frac{\Gamma \vdash \mathcal{M}_1 : R \quad \cdots \quad \Gamma \vdash \mathcal{M}_\ell : R}{\Gamma \vdash \underline{f}(\mathcal{M}_1, \ldots, \mathcal{M}_\ell) : R}$$

$$\frac{\Gamma, y : \tau \vdash \mathcal{M} : \tau'}{\Gamma \vdash \lambda y.\, \mathcal{M} : \tau \to \tau'} \qquad \frac{\Gamma \vdash \mathcal{M} : \tau \to \tau' \quad \Gamma \vdash \mathcal{N} : \tau}{\Gamma \vdash \mathcal{M}\, \mathcal{N} : \tau'} \qquad \frac{\Gamma \vdash \mathcal{M} : (\tau \to \tau') \to \tau \to \tau'}{\Gamma \vdash \mathbf{Y}\mathcal{M} : \tau \to \tau'}$$

$$\frac{\Gamma \vdash \mathcal{L} : R \quad \Gamma \vdash \mathcal{M} : \tau \quad \Gamma \vdash \mathcal{N} : \tau}{\Gamma \vdash \mathbf{if}\ \mathcal{L} < 0\, \mathbf{then}\, \mathcal{M}\, \mathbf{else}\, \mathcal{N} : \tau} \qquad \frac{}{\Gamma \vdash \mathbf{sample} : R} \qquad \frac{\Gamma \vdash \mathcal{M} : R}{\Gamma \vdash \mathbf{score}(\mathcal{M}) : R}$$

(b) Type system for symbolic terms

$$\mathcal{R} ::= (\lambda y.\, \mathcal{M})\, \mathcal{V} \mid \underline{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell) \mid \mathbf{if}\ \mathcal{V} < 0\, \mathbf{then}\, \mathcal{M}\, \mathbf{else}\, \mathcal{N} \mid \mathbf{Y}(\lambda y.\, \mathcal{M})$$
$$\mid \mathbf{sample} \mid \mathbf{score}(\mathcal{V})$$
$$\mathcal{E} ::= [\,] \mid \mathcal{E}\, \mathcal{M} \mid (\lambda y.\, \mathcal{M})\, \mathcal{E} \mid \underline{f}(\mathcal{V}_1, \ldots, \mathcal{V}_{i-1}, \mathcal{E}, \mathcal{M}_{i+1}, \ldots, \mathcal{M}_\ell)$$
$$\mid \mathbf{if}\ \mathcal{E} < 0\, \mathbf{then}\, \mathcal{M}\, \mathbf{else}\, \mathcal{N} \mid \mathbf{Y}\mathcal{E} \mid \mathbf{score}(\mathcal{E})$$

(c) Symbolic values (typically $\mathcal{V}$), redexes ($\mathcal{R}$) and reduction contexts ($\mathcal{E}$).

Figure 3.1: Symbolic terms and values, type system, reduction contexts, and redexes. As usual $f \in \mathrm{Op}$ and $r \in \mathbb{R}$.

Each symbolic term $\mathcal{M} \in \Lambda_{(m,n)}$ has a corresponding set of regular terms, accounting for all possible values for its sampling variables $\alpha_1, \ldots, \alpha_n$ and its (other) free variables $\theta_1, \ldots, \theta_m$. For $\boldsymbol{r} \in \mathbb{R}^m$ and $\boldsymbol{s} \in \mathbb{S}_n$, we call ***partially evaluated instantiation*** of $\mathcal{M}$ the term $\lfloor \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s})$ obtained from $\mathcal{M}[\underline{\boldsymbol{r}}/\boldsymbol{\theta}, \underline{\boldsymbol{s}}/\boldsymbol{\alpha}]$ by recursively "evaluating" subterms of the form $\boxed{f}(\underline{r_1}, \ldots, \underline{r_\ell})$ to $\underline{f(r_1, \ldots, r_\ell)}$, provided $(r_1, \ldots, r_\ell) \in \operatorname{dom} f$. In this operation, subterms of the form $\underline{f}(\underline{r_1}, \ldots, \underline{r_\ell})$ are left unchanged, and so are any other redexes. $\lfloor \mathcal{M} \rfloor$ can be viewed as a partial function $\lfloor \mathcal{M} \rfloor : \mathbb{R}^m \times \mathbb{S}_n \rightharpoonup \Lambda$ and a formal definition[5] is presented in Fig. 3.2b. Observe that for $\mathcal{M} \in \Lambda_{(m,n)}$ and $(\boldsymbol{r}, \boldsymbol{s}) \in \operatorname{dom} \lfloor \mathcal{M} \rfloor$, $\lfloor \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s})$ is a closed term.

Example 3.6. Consider $\mathcal{M} \equiv (\lambda z.\, \alpha_1 \boxdot \underline{3})\, (\mathbf{score}(\mathrm{pdf}_{\mathcal{N}(1.1,0.1)}(\alpha_2)))$. Then, for $\boldsymbol{r} = []$ and $\boldsymbol{s} = [0.2, 0.9, 0.7]$, we have $\lfloor \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s}) = (\lambda z.\, \underline{0.6})\, (\mathbf{score}(\underline{\mathrm{pdf}_{\mathcal{N}(1.1,0.1)}}(\underline{0.9})))$.

More generally, observe that if $\Gamma \vdash \mathcal{M} : \tau$ and $(\boldsymbol{r}, \boldsymbol{s}) \in \operatorname{dom} \lfloor \mathcal{M} \rfloor$ then $\Gamma \vdash \lfloor \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s}) : \tau$. In order to evaluate conditionals $\mathbf{if}\ \mathcal{L} < 0\ \mathbf{then}\ \mathcal{M}\ \mathbf{else}\ \mathcal{N}$ we need to reduce $\mathcal{L}$ to a real constant, i.e., we need to have $\lfloor \mathcal{L} \rfloor (\boldsymbol{r}, \boldsymbol{s}) = \underline{r}$ for some $r \in \mathbb{R}$. This is the case whenever $\mathcal{L}$ is a symbolic value of type R, since these are built only out of delayed operations, real constants and distinguished variables θ_i or α_j. Indeed, we can show the following:

Lemma 3.7. *Let $(\boldsymbol{r}, \boldsymbol{s}) \in \operatorname{dom} \lfloor \mathcal{M} \rfloor$. Then $\mathcal{M}$ is a symbolic value iff $\lfloor \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s})$ is a value.*

Proof sketch. First, suppose $\mathcal{M}$ is a symbolic value $\mathcal{V}$. It is easy to prove inductively that for a symbolic value $\mathcal{V}$ of type R, $\lfloor \mathcal{V} \rfloor (\boldsymbol{r}, \boldsymbol{s})$ is a real constant. Otherwise, both $\mathcal{V}$ and $\lfloor \mathcal{V} \rfloor (\boldsymbol{r}, \boldsymbol{s})$ are abstractions.

Conversely, suppose $\lfloor \mathcal{V} \rfloor (\boldsymbol{r}, \boldsymbol{s})$ is a value. If it is an abstraction then so is $\mathcal{V}$. Otherwise, it is a real constant. By the definition of $\lfloor \cdot \rfloor$ and a case inspection of $\mathcal{M}$ this is only possible if $\mathcal{M}$ is a symbolic value. □

For symbolic values $\mathcal{V} : R$ and $(\boldsymbol{r}, \boldsymbol{s}) \in \operatorname{dom} \lfloor \mathcal{V} \rfloor$ we employ the notation $\|\mathcal{V}\| (\boldsymbol{r}, \boldsymbol{s}) := r'$ provided that $\lfloor \mathcal{V} \rfloor (\boldsymbol{r}, \boldsymbol{s}) = \underline{r'}$.

A simple induction on symbolic terms and values (see Appendix A.2.1) yields the following property, which is crucial for the proof of our main result (Theorem 3.18):

Lemma 3.8. *Suppose the set* Op *of primitives satisfies Assumption 3.1(i).*

(i) *For each symbolic value $\mathcal{V}$ of type R, by identifying* $\operatorname{dom} \|\mathcal{V}\|$ *with a subset of $\mathbb{R}^{m+n}$, we have $\|\mathcal{V}\| \in$ Op.*

(ii) *If* Op *also satisfies Assumption 3.1(ii) then for each symbolic term $\mathcal{M}$,*

$$\lfloor \mathcal{M} \rfloor : \mathbb{R}^m \times \mathbb{S}_n \rightharpoonup \Lambda$$

is differentiable in the interior of its domain.

3.2.2 Symbolic Operational Semantics

We aim to develop a symbolic operational semantics that provides a sound and complete abstraction of the (concrete) operational trace semantics. The symbolic semantics

[5] To be completely rigorous, we define for *fixed* m and n, partial functions $\lfloor \mathcal{M} \rfloor_{m,n} : \mathbb{R}^m \times \mathbb{S}_n \rightharpoonup \Lambda$ for symbolic terms $\mathcal{M}$ whose distinguished variables are amongst $\theta_1, \ldots, \theta_m$ and $\alpha_1, \ldots, \alpha_n$. $\mathcal{M}$ may contain other variables $y, z, \ldots$ of any type. Since m and n are usually clear from the context, we omit them.

$$
\begin{aligned}
\operatorname{dom}\left\lfloor \boxed{f}(\mathcal{V}_1,\ldots,\mathcal{V}_\ell)\right\rfloor &\coloneqq \{(\boldsymbol{r},\boldsymbol{s}) \in \operatorname{dom}\lfloor\mathcal{V}_1\rfloor \cap \cdots \cap \operatorname{dom}\lfloor\mathcal{V}_\ell\rfloor \mid \\
&\qquad (r'_1,\ldots,r'_\ell) \in \operatorname{dom} f, \text{where} \\
&\qquad \underline{r'_1} = \lfloor\mathcal{V}_1\rfloor(\boldsymbol{r},\boldsymbol{s}),\cdots,\underline{r'_\ell} = \lfloor\mathcal{V}_\ell\rfloor(\boldsymbol{r},\boldsymbol{s})\} \\
\operatorname{dom}\lfloor\mathbf{sample}\rfloor &\coloneqq \operatorname{dom}\lfloor\theta_i\rfloor \coloneqq \operatorname{dom}\lfloor\alpha_j\rfloor \\
&\coloneqq \operatorname{dom}\lfloor y\rfloor \coloneqq \operatorname{dom}\left\lfloor \underline{r'}\right\rfloor \coloneqq \mathbb{R}^m \times \mathbb{S}_n \\
\operatorname{dom}\underline{f}(\mathcal{M}_1,\ldots,\mathcal{M}_\ell) &\coloneqq \operatorname{dom}\lfloor\mathcal{M}_1\rfloor \cap \cdots \cap \operatorname{dom}\lfloor\mathcal{M}_\ell\rfloor \\
\operatorname{dom}\lfloor\lambda y.\,\mathcal{M}\rfloor &\coloneqq \operatorname{dom}\lfloor\mathbf{Y}\,\mathcal{M}\rfloor \coloneqq \operatorname{dom}\lfloor\mathbf{score}(\mathcal{M})\rfloor \coloneqq \operatorname{dom}\lfloor\mathcal{M}\rfloor \\
\operatorname{dom}\lfloor\mathcal{M}\,\mathcal{N}\rfloor &\coloneqq \operatorname{dom}\lfloor\mathcal{M}\rfloor \cap \operatorname{dom}\lfloor\mathcal{N}\rfloor \\
\operatorname{dom}\lfloor\mathbf{if}\,\mathcal{L}<0\,\mathbf{then}\,\mathcal{M}\,\mathbf{else}\,\mathcal{N}\rfloor &\coloneqq \operatorname{dom}\lfloor\mathcal{L}\rfloor \cap \operatorname{dom}\lfloor\mathcal{M}\rfloor \cap \operatorname{dom}\lfloor\mathcal{N}\rfloor
\end{aligned}
$$

(a) Domain of $\lfloor\cdot\rfloor$

$$
\begin{aligned}
\left\lfloor \boxed{f}(\mathcal{V}_1,\ldots,\mathcal{V}_\ell)\right\rfloor(\boldsymbol{r},\boldsymbol{s}) &\coloneqq \underline{f(r'_1,\ldots,r'_\ell)}\text{, where for } 1\le i\le \ell,\ \lfloor\mathcal{V}_i\rfloor(\boldsymbol{r},\boldsymbol{s}) = \underline{r'_i} \\
\lfloor\theta_i\rfloor(\boldsymbol{r},\boldsymbol{s}) &\coloneqq \underline{r_i} \\
\lfloor\alpha_j\rfloor(\boldsymbol{r},\boldsymbol{s}) &\coloneqq \underline{s_j} \\
\lfloor y\rfloor(\boldsymbol{r},\boldsymbol{s}) &\coloneqq y \\
\left\lfloor \underline{r'}\right\rfloor(\boldsymbol{r},\boldsymbol{s}) &\coloneqq \underline{r'} \\
\left\lfloor \underline{f}(\mathcal{M}_1,\ldots,\mathcal{M}_\ell)\right\rfloor(\boldsymbol{r},\boldsymbol{s}) &\coloneqq \underline{f}(\lfloor\mathcal{M}_1\rfloor(\boldsymbol{r},\boldsymbol{s}),\ldots,\lfloor\mathcal{M}_\ell\rfloor(\boldsymbol{r},\boldsymbol{s})) \\
\lfloor\lambda y.\,\mathcal{M}\rfloor(\boldsymbol{r},\boldsymbol{s}) &\coloneqq \lambda y.\,\lfloor\mathcal{M}\rfloor(\boldsymbol{r},\boldsymbol{s}) \\
\lfloor\mathcal{M}\,\mathcal{N}\rfloor(\boldsymbol{r},\boldsymbol{s}) &\coloneqq (\lfloor\mathcal{M}\rfloor(\boldsymbol{r},\boldsymbol{s}))\,(\lfloor\mathcal{N}\rfloor(\boldsymbol{r},\boldsymbol{s})) \\
\lfloor\mathbf{Y}\,\mathcal{M}\rfloor(\boldsymbol{r},\boldsymbol{s}) &\coloneqq \mathbf{Y}(\lfloor\mathcal{M}\rfloor(\boldsymbol{r},\boldsymbol{s})) \\
\lfloor\mathbf{if}\,\mathcal{L}<0\,\mathbf{then}\,\mathcal{M}\,\mathbf{else}\,\mathcal{N}\rfloor(\boldsymbol{r},\boldsymbol{s}) &\coloneqq \mathbf{if}\ \lfloor\mathcal{L}\rfloor(\boldsymbol{r},\boldsymbol{s})<0\,\mathbf{then}\ \lfloor\mathcal{M}\rfloor(\boldsymbol{r},\boldsymbol{s})\,\mathbf{else}\ \lfloor\mathcal{N}\rfloor(\boldsymbol{r},\boldsymbol{s}) \\
\lfloor\mathbf{sample}\rfloor(\boldsymbol{r},\boldsymbol{s}) &\coloneqq \mathbf{sample} \\
\lfloor\mathbf{score}(\mathcal{M})\rfloor(\boldsymbol{r},\boldsymbol{s}) &\coloneqq \mathbf{score}(\lfloor\mathcal{M}\rfloor(\boldsymbol{r},\boldsymbol{s}))
\end{aligned}
$$

(b) Definition of $\lfloor\cdot\rfloor$ on $\operatorname{dom}\lfloor\cdot\rfloor$

Figure 3.2: Formal definition of the instantiation and partial evaluation function $\lfloor\cdot\rfloor$

is presented as a rewrite system of ***symbolic configurations***, which are defined to be triples of the form $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle$, where for some m and n, $\mathcal{M} \in \Lambda_{(m,n)}$, the set $U \subseteq \operatorname{dom} \lfloor \mathcal{M} \rfloor \subseteq \mathbb{R}^m \times \mathbb{S}_n$ is measurable, and $w : \mathbb{R}^m \times \mathbb{S} \rightharpoonup \mathbb{R}_{\geq 0}$ with $\operatorname{dom} w = U$. Thus, we aim to prove the following result (writing 1 for the constant function $\mathbb{1}(\boldsymbol{r}, \boldsymbol{s}).\, 1$):

Theorem 3.9. *Let M be a term with free variables amongst $\theta_1, \ldots, \theta_m$.*

(i) (Soundness). *If $\langle\!\langle M, 1, \mathbb{R}^m \rangle\!\rangle \Rightarrow^* \langle\!\langle \mathcal{V}, w, U \rangle\!\rangle$ then for all $(\boldsymbol{r}, \boldsymbol{s}) \in U$,*

$$\operatorname{weight}_M(\boldsymbol{r}, \boldsymbol{s}) = w(\boldsymbol{r}, \boldsymbol{s}) \qquad \operatorname{value}_M(\boldsymbol{r}, \boldsymbol{s}) = \lfloor \mathcal{V} \rfloor (\boldsymbol{r}, \boldsymbol{s})$$

(ii) (Completeness). *If $\boldsymbol{r} \in \mathbb{R}^m$ and $\langle M[\underline{\boldsymbol{r}}/\boldsymbol{\theta}], 1, [] \rangle \to^* \langle V, w, \boldsymbol{s} \rangle$ then there exists $\langle\!\langle M, 1, \mathbb{R}^m \rangle\!\rangle \Rightarrow^* \langle\!\langle \mathcal{V}, w, U \rangle\!\rangle$ such that $(\boldsymbol{r}, \boldsymbol{s}) \in U$.*

As formalised by Theorem 3.9, the key intuition behind symbolic configurations $\langle\!\langle \mathcal{V}, w, U \rangle\!\rangle$ (that are reachable from a given $\langle\!\langle M, 1, \mathbb{R}^m \rangle\!\rangle$) is that:

– $\mathcal{V}$ gives a correct *local* view of value_M (restricted to U), and

– w gives a correct *local* view of weight_M (restricted to U);

moreover, the respective third components U (of the symbolic configurations $\langle\!\langle \mathcal{V}, w, U \rangle\!\rangle$) cover $\mathbb{T}_{M,\mathrm{term}}$.

To establish Theorem 3.9, we introduce ***symbolic reduction contexts*** and ***symbolic redexes***. These are presented in Fig. 3.1c and extend the usual notions (replacing real constants with arbitrary symbolic values of type R).

Using Lemma 3.7 we obtain:

Lemma 3.10. *If $\mathcal{R}$ is a symbolic redex and $(\boldsymbol{r}, \boldsymbol{s}) \in \operatorname{dom} \lfloor \mathcal{R} \rfloor$ then $\lfloor \mathcal{R} \rfloor (\boldsymbol{r}, \boldsymbol{s})$ is a redex.*

The following can be proven by a straightforward induction (see Appendix A.2.2):

Lemma 3.11 (Subject Construction)**.** *Let $\mathcal{M}$ be a symbolic term.*

(i) *If $\mathcal{M}$ is a symbolic value then for all symbolic contexts $\mathcal{E}$ and symbolic redexes $\mathcal{R}$, $\mathcal{M} \not\equiv \mathcal{E}[\mathcal{R}]$.*
(ii) *If $\mathcal{M} \equiv \mathcal{E}_1[\mathcal{R}_1] \equiv \mathcal{E}_2[\mathcal{R}_2]$ then $\mathcal{E}_1 \equiv \mathcal{E}_2$ and $\mathcal{R}_1 \equiv \mathcal{R}_2$.*
(iii) *If $\mathcal{M}$ is not a symbolic value and $\operatorname{dom} \lfloor \mathcal{M} \rfloor \neq \emptyset$ then there exist $\mathcal{E}$ and $\mathcal{R}$ such that $\mathcal{M} \equiv \mathcal{E}[\mathcal{R}]$.*

The partial instantiation function also extends to symbolic contexts $\mathcal{E}$ in the evident way – we give the full definition in Appendix A.2.2 (Definition A.3).

Now, we introduce the following rules for ***symbolic redex contractions***:

$$\langle\!\langle (\lambda y.\, \mathcal{M})\, \mathcal{V}, w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{M}[\mathcal{V}/y], w, U \rangle\!\rangle$$

$$\langle\!\langle \underline{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell), w, U \rangle\!\rangle \Rightarrow \langle\!\langle \boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell), w, \operatorname{dom} \|\boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell)\| \cap U \rangle\!\rangle$$

$$\langle\!\langle \mathbf{Y}(\lambda y.\, \mathcal{M}), w, U \rangle\!\rangle \Rightarrow \langle\!\langle \lambda z.\, \mathcal{M}\, [\mathbf{Y}(\lambda y.\, \mathcal{M})/y]\, z, w, U \rangle\!\rangle$$

$$\langle\!\langle \mathbf{if}\, \mathcal{V} < 0\, \mathbf{then}\, \mathcal{M}\, \mathbf{else}\, \mathcal{N}, w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{M}, w, \|\mathcal{V}\|^{-1}\, (-\infty, 0) \cap U \rangle\!\rangle$$

$$\langle\!\langle \mathbf{if}\, \mathcal{V} < 0\, \mathbf{then}\, \mathcal{M}\, \mathbf{else}\, \mathcal{N}, w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{N}, w, \|\mathcal{V}\|^{-1}\, [0, \infty) \cap U \rangle\!\rangle$$

$$\langle\!\langle \mathbf{sample}, w, U \rangle\!\rangle \Rightarrow \langle\!\langle \alpha_{n+1}, w', U' \rangle\!\rangle \qquad (U \subseteq \mathbb{R}^m \times \mathbb{S}_n)$$

$$\langle\!\langle \mathbf{score}(\mathcal{V}), w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{V}, \|\mathcal{V}\| \cdot w, \|\mathcal{V}\|^{-1}[0,\infty) \cap U \rangle\!\rangle$$

In the rule for **sample** (recall that in this chapter we focus on the uniform distribution on $(0,1)$), $U' \coloneqq \{(\boldsymbol{r}, \boldsymbol{s} +\!\!+ [s']) \mid (\boldsymbol{r}, \boldsymbol{s}) \in U \wedge s' \in (0,1)\}$ and $w'(\boldsymbol{r}, \boldsymbol{s} +\!\!+ [s']) \coloneqq w(\boldsymbol{r}, \boldsymbol{s})$; in the rule for $\mathbf{score}(\mathcal{V})$, $(\|\mathcal{V}\| \cdot w)(\boldsymbol{r}, \boldsymbol{s}) \coloneqq \|\mathcal{V}\|(\boldsymbol{r}, \boldsymbol{s}) \cdot w(\boldsymbol{r}, \boldsymbol{s})$.

The rules are designed to closely mirror their concrete counterparts. Crucially, the rule for **sample** introduces a "fresh" sampling variable, and the two rules for conditionals split the last component $U \subseteq \mathbb{R}^m \times \mathbb{S}_n$ according to whether $\|\mathcal{V}\|(\boldsymbol{r}, \boldsymbol{s}) < 0$ or $\|\mathcal{V}\|(\boldsymbol{r}, \boldsymbol{s}) \geq 0$. The "delay" contraction (second rule) is introduced for a technical reason: ultimately, to enable Theorem 3.9(i) (Soundness). Otherwise, it is, for example, unclear whether $\lambda y.\, \alpha_1 + \underline{1}$ should correspond to $\lambda y.\, \underline{0.5} + \underline{1}$ or $\lambda y.\, \underline{1.5}$ for $s_1 = 0.5$.

Finally, we lift this to arbitrary symbolic terms using the obvious rule for symbolic evaluation contexts:

$$\frac{\langle\!\langle \mathcal{R}, w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{R}', w', U' \rangle\!\rangle}{\langle\!\langle \mathcal{E}[\mathcal{R}], w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{E}[\mathcal{R}'], w', U' \rangle\!\rangle}$$

Note that we do not need rules corresponding to reductions to fail because the third component of the symbolic configurations "filters out" the pairs $(\boldsymbol{r}, \boldsymbol{s})$ corresponding to undefined behaviour. In particular, the following holds:

Lemma 3.12. *Suppose $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle$ is a symbolic configuration and*

$$\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{N}, w', U' \rangle\!\rangle$$

Then $\langle\!\langle \mathcal{N}, w', U' \rangle\!\rangle$ is a symbolic configuration.

Proof. Suppose that $\mathcal{M} \equiv \mathcal{E}[\mathcal{R}]$ and $\mathcal{N} \equiv \mathcal{E}[\mathcal{R}']$. Because of Lemma 3.8 and the assumption that the functions in Op are measurable, U' is measurable again. Furthermore, the rules ensure that $U' \subseteq \operatorname{dom} \lfloor \mathcal{R}' \rfloor$. (For the first rule this is because of the Substitution Eq. (A.1).) By the Substitution Eq. (A.3),

$$U' \subseteq \operatorname{dom} \lfloor \mathcal{E}[\mathcal{R}] \rfloor \cap \operatorname{dom} \lfloor \mathcal{R}' \rfloor \subseteq \operatorname{dom} \lfloor \mathcal{E} \rfloor \cap \operatorname{dom} \lfloor \mathcal{R}' \rfloor = \operatorname{dom} \lfloor \mathcal{E}[\mathcal{R}'] \rfloor \qquad \square$$

A key advantage of the symbolic execution is that the induced computation tree is finitely branching, since branching only arises from conditionals, splitting the trace space into disjoint subsets. This contrasts with the concrete situation (from Section 2.2.2), in which sampling creates uncountably many branches.

Lemma 3.13 (Basic Properties)**.** *Let $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle$ be a symbolic configuration. Then*

(i) *There are at most countably distinct $\mathcal{N}$, w' and U' s.t. $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle \Rightarrow^* \langle\!\langle \mathcal{N}, w', U' \rangle\!\rangle$.*
(ii) *If $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle \Rightarrow^* \langle\!\langle \mathcal{V}_i, w_i, U_i \rangle\!\rangle$ for $i \in \{1,2\}$ then $U_1 = U_2$ or $U_1 \cap U_2 = \emptyset$.*
(iii) *If $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle \Rightarrow^* \langle\!\langle \mathcal{E}_i[\mathbf{sample}], w_i, U_i \rangle\!\rangle$ for $i \in \{1,2\}$ then $U_1 = U_2$ or $U_1 \cap U_2 = \emptyset$.*

Proof sketch. By subject construction (Lemma 3.11), there is at most one $\mathcal{E}$ and $\mathcal{R}$ such that $\mathcal{M} \equiv \mathcal{E}[\mathcal{R}]$. An inspection of the rules shows that U' such that

$$\langle\!\langle \mathcal{R}, w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{R}', w', U' \rangle\!\rangle$$

is unique unless $\mathcal{R}$ is a conditional, in which case there are two distinct such U'. Hence, there are at most two distinct U' such that $\langle\!\langle \mathcal{E}[\mathcal{R}], w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{N}, w', U' \rangle\!\rangle$. The first part follows by induction on the number of reduction steps.

For the other two parts note that if $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{N}, w', U' \rangle\!\rangle$ either $U' \subseteq U$ or $U' = \{(\boldsymbol{r}, \boldsymbol{s} +\!\!+ [r']) \mid (\boldsymbol{r}, \boldsymbol{s}) \in U \wedge r' \in (0,1)\}$. In particular, if $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle \Rightarrow^* \langle\!\langle \mathcal{N}, w', U' \rangle\!\rangle$, $U' \subseteq \{(\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) \mid (\boldsymbol{r}, \boldsymbol{s}) \in U \wedge \boldsymbol{s'} \in \mathbb{S}_n\}$ for some $n \in \mathbb{N}$.

By the discussion for the first part of the lemma, if $\langle \mathcal{M}, w, U \rangle \Rightarrow^* \langle\!\langle \mathcal{N}_i, w_i, U_i \rangle\!\rangle$ for $i \in \{1,2\}$, then w.l.o.g., either

(i) $\langle\!\langle \mathcal{N}_1, w_1, U_1 \rangle\!\rangle \Rightarrow^* \langle\!\langle \mathcal{N}_2, w_2, U_2 \rangle\!\rangle$ or
(ii) $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle \Rightarrow^* \langle\!\langle \mathcal{E}[\mathbf{if}\ \mathcal{L} < 0\ \mathbf{then}\ \mathcal{M}_1\ \mathbf{else}\ \mathcal{M}_2], w', U' \rangle\!\rangle$ and

$$\langle\!\langle \mathcal{E}[\mathbf{if}\ \mathcal{L} < 0\ \mathbf{then}\ \mathcal{M}_1\ \mathbf{else}\ \mathcal{M}_2], w', U' \rangle\!\rangle \overset{\nearrow^*}{\searrow^*} \begin{array}{c} \langle\!\langle \mathcal{E}[\mathcal{M}_1], w', U' \cap \|\mathcal{L}\|^{-1}(-\infty, 0) \rangle\!\rangle \\ \Rightarrow^* \langle\!\langle \mathcal{N}_1, w_1, U_1 \rangle\!\rangle \\ \langle\!\langle \mathcal{E}[\mathcal{M}_2], w', U' \cap \|\mathcal{L}\|^{-1}[0, \infty) \rangle\!\rangle \\ \Rightarrow^* \langle\!\langle \mathcal{N}_2, w_2, U_2 \rangle\!\rangle \end{array}$$

for suitable $\mathcal{N}$, $\mathcal{E}$, $\mathcal{L}$, $\mathcal{M}_1$, $\mathcal{M}_2$, w' and U'.

In the latter case in particular $U_1 \cap U_2 = \emptyset$ holds.

This implies the second and third part of the lemma. □

Crucially, there is a correspondence between the concrete and symbolic semantics in that they can "simulate" each other:

Proposition 3.14 (Correspondence)**.** *Suppose $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle$ is a symbolic configuration, and $(\boldsymbol{r}, \boldsymbol{s}) \in U$. Let $M \equiv \lfloor \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s})$ and $w := w(\boldsymbol{r}, \boldsymbol{s})$. Then*

(i) *If $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{N}, w', U' \rangle\!\rangle$ and $(\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) \in U'$ then*

$$\langle M, w, \boldsymbol{s} \rangle \to \left\langle \lfloor \mathcal{N} \rfloor (\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}), w(\boldsymbol{r}, \boldsymbol{s'}), \boldsymbol{s} +\!\!+ \boldsymbol{s'} \right\rangle.$$

(ii) *If $\langle M, w, \boldsymbol{s} \rangle \to \langle N, w', \boldsymbol{s'} \rangle$ then there exists $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{N}, w', U' \rangle\!\rangle$ such that $\lfloor \mathcal{N} \rfloor (\boldsymbol{r}, \boldsymbol{s'}) \equiv N$, $w'(\boldsymbol{r}, \boldsymbol{s'}) = w'$ and $(\boldsymbol{r}, \boldsymbol{s'}) \in U'$.*

Proof. Suppose that $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle$ is a symbolic configuration and $(\boldsymbol{r}, \boldsymbol{s}) \in U$.

If $\mathcal{M}$ is a symbolic value then $\lfloor \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s})$ is a value Lemma 3.7 and there is nothing to prove.

Otherwise, by Lemma 3.11, there exists unique $\mathcal{E}$ and $\mathcal{R}$ such that $\mathcal{M} \equiv \mathcal{E}[\mathcal{R}]$. Thus, we can define the context $E \equiv \lfloor \mathcal{E} \rfloor (\boldsymbol{r}, \boldsymbol{s})$ and redex $R \equiv \lfloor \mathcal{R} \rfloor (\boldsymbol{r}, \boldsymbol{s})$ (see Lemma 3.10), and it holds by Eq. (A.3), $\lfloor \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s}) \equiv E[R]$.

(i) If $\langle\!\langle \mathcal{R}, w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{R}', w', U' \rangle\!\rangle$ and $(\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) \in U'$ then by case inspection (see Lemma A.1),

$$\langle R, w(\boldsymbol{r}, \boldsymbol{s}), \boldsymbol{s} \rangle \to \left\langle R', w'(\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}), \boldsymbol{s} +\!\!+ \boldsymbol{s'} \right\rangle$$

such that $R' \equiv \lfloor \mathcal{R}' \rfloor (\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'})$. So,

$$\langle E[R], w(\boldsymbol{r}, \boldsymbol{s}), \boldsymbol{s} \rangle \to \left\langle E[R'], w'(\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}), \boldsymbol{s} +\!\!+ \boldsymbol{s'} \right\rangle$$

and by the Substitution Eq. (A.3), $E[R'] \equiv \lfloor \mathcal{E}[\mathcal{R}'] \rfloor (\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'})$.

(ii) Conversely, if

$$\langle R, w(\boldsymbol{r}, \boldsymbol{s}), \boldsymbol{s} \rangle \to \langle R', w', \boldsymbol{s'} \rangle$$

then a simple case analysis (see Lemma A.2) shows that for some $\mathcal{R}'$, w' and U',

$$\langle\!\langle \mathcal{R}, w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{R}', w', U' \rangle\!\rangle$$

such that $\lfloor \mathcal{R}' \rfloor (\boldsymbol{r}, \boldsymbol{s'}) \equiv R'$, $w(\boldsymbol{r}, \boldsymbol{s'}) = w'$ and $(\boldsymbol{r}, \boldsymbol{s'}) \in U'$. Thus, also

$$\langle\!\langle \mathcal{E}[\mathcal{R}], w, U \rangle\!\rangle \Rightarrow^* \langle\!\langle \mathcal{E}[\mathcal{R}'], w', U' \rangle\!\rangle$$

and by the Substitution Eq. (A.3), we have $\lfloor \mathcal{E}[\mathcal{R}'] \rfloor (\boldsymbol{r}, \boldsymbol{s}) \equiv E[R]$. □

3.3 Densities of Almost Surely Terminating Programs are Differentiable Almost Everywhere

So far we have seen that the symbolic execution semantics provides a sound and complete way to reason about the weight and value functions. In this section we impose further restrictions on the primitive operations and the terms to obtain results about the differentiability of these functions.

Henceforth, we assume Assumption 3.1 and we fix a term M with free variables amongst $\theta_1, \ldots, \theta_m$.

From Lemma 3.8 we immediately obtain the following[6]:

Lemma 3.15. *Let $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle$ be a symbolic configuration such that w is differentiable on $\mathring{U}$ and $\mu(\partial U) = 0$. If $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{M}', w', U' \rangle\!\rangle$ then w' is differentiable on $\mathring{U}'$ and $\mu(\partial U') = 0$.*

Proof. For the **Score**-rule this is due to Lemma 3.8 and the fact that differentiable functions are closed under multiplication. For the other rules differentiability of w' is obvious.

Furthermore, note that $\mu(\partial\{(\boldsymbol{r}, \boldsymbol{s} +\!\!+ [s']) \mid (\boldsymbol{r}, \boldsymbol{s}) \in U \wedge s' \in (0,1)\}) = \mu(\partial U)$ and for symbolic values $\mathcal{V}$,

$$\mu\left(\partial\left(\|\mathcal{V}\|^{-1}(-\infty, 0)\right)\right) = \mu\left(\partial\left(\|\mathcal{V}\|^{-1}[0, \infty)\right)\right) = 0$$

because of Lemma 3.8 and Assumption 3.1. Consequently, due to the general fact that $\partial(U \cap V) \subseteq \partial U \cup \partial V$, in any case, $\mu(\partial U') = 0$. □

3.3.1 Differentiability on Terminating Traces

As an immediate consequence of the preceding, Lemma 3.8 and the Soundness Theorem 3.9(i), whenever $\langle\!\langle M, 1, \mathbb{R}^m \rangle\!\rangle \Rightarrow^* \langle\!\langle \mathcal{V}, w, U \rangle\!\rangle$ then weight_M and value_M are differentiable everywhere in $\mathring{U}$.

Recall the set $\mathbb{T}_{M,\mathrm{term}}$ of $(\boldsymbol{r}, \boldsymbol{s}) \in \mathbb{R}^m \times \mathbb{S}$ from Eq. (3.1) for which M terminates. We abbreviate $\mathbb{T}_{M,\mathrm{term}}$ to $\mathbb{T}_{\mathrm{term}}$ and define

$$\mathbb{T}_{\mathrm{term}} \coloneqq \mathbb{T}_{M,\mathrm{term}} = \{(\boldsymbol{r}, \boldsymbol{s}) \in \mathbb{R}^m \times \mathbb{S} \mid \exists V, w\,.\, \langle M[\boldsymbol{r}/\boldsymbol{\theta}], 1, [] \rangle \to^* \langle V, w, \boldsymbol{s} \rangle\}$$

[6] Henceforth, we abbreviate $\mu_{\mathbb{R}^m \times \mathbb{S}}$ from Section 2.2.2 to μ to reduce clutter.

$$\mathbb{T}^{\mathrm{int}}_{\mathrm{term}} \coloneqq \bigcup \{\mathring{U} \mid \exists \mathcal{V}, \boldsymbol{w} \,.\, \langle\!\langle M, \mathit{1}, \mathbb{R}^m \rangle\!\rangle \Rightarrow^* \langle\!\langle \mathcal{V}, \boldsymbol{w}, U \rangle\!\rangle\}$$

By Completeness Theorem 3.9(ii),

$$\mathbb{T}_{\mathrm{term}} = \bigcup \{U \mid \exists \mathcal{V}, \boldsymbol{w} \,.\, \langle\!\langle M, \mathit{1}, \mathbb{R}^m \rangle\!\rangle \Rightarrow^* \langle\!\langle \mathcal{V}, \boldsymbol{w}, U \rangle\!\rangle\}$$

Therefore, being countable unions of measurable sets (Lemmas 3.12 and 3.13), $\mathbb{T}_{\mathrm{term}}$ and $\mathbb{T}^{\mathrm{int}}_{\mathrm{term}}$ are measurable.

By what we have said above, weight_M and value_M are differentiable everywhere on $\mathbb{T}^{\mathrm{int}}_{\mathrm{term}}$. Observe that in general, $\mathbb{T}^{\mathrm{int}}_{\mathrm{term}} \subsetneq \mathbb{T}^{\circ}_{\mathrm{term}}$. However,

$$\mu\left(\mathbb{T}_{\mathrm{term}} \setminus \mathbb{T}^{\mathrm{int}}_{\mathrm{term}}\right) = \mu\Bigg(\bigcup_{\substack{U : \langle\!\langle M, \mathit{1}, \mathbb{R}^m \rangle\!\rangle \Rightarrow^* \\ \langle\!\langle \mathcal{V}, \boldsymbol{w}, U \rangle\!\rangle}} \left(U \setminus \mathring{U}\right)\Bigg) \leq \sum_{\substack{U : \langle\!\langle M, \mathit{1}, \mathbb{R}^m \rangle\!\rangle \Rightarrow^* \\ \langle\!\langle \mathcal{V}, \boldsymbol{w}, U \rangle\!\rangle}} \mu(\partial U) = 0 \tag{3.2}$$

The first equation holds because the U-indexed union is of pairwise disjoint sets. The inequality is due to $(U \setminus \mathring{U}) \subseteq \partial U$. The last equation above holds because each $\mu(\partial U) = 0$ (Assumption 3.1 and Lemma 3.15).

Thus, we conclude:

Theorem 3.16. *Let M be an SPCF term. Then its weight function* weight_M *and value function* value_M *are differentiable for almost all terminating traces.*

3.3.2 Differentiability for Almost Surely Terminating Terms

Next, we would like to extend this insight for almost surely terminating terms to suitable subsets of $\mathbb{R}^m \times \mathbb{S}$, the union of which constitutes almost the entirety of $\mathbb{R}^m \times \mathbb{S}$. Therefore, it is worth examining consequences of almost sure termination (see Definition 3.4).

We say that $(\boldsymbol{r}, \boldsymbol{s}) \in \mathbb{R}^m \times \mathbb{S}$ is *maximal* (for M) if $\langle M[\underline{\boldsymbol{r}}/\boldsymbol{\theta}], 1, []\rangle \to^* \langle N, w, \boldsymbol{s}\rangle$ and for all $\boldsymbol{s'} \in \mathbb{S} \setminus \{[]\}$ and N', $\langle N, w, \boldsymbol{s}\rangle \not\to^* \langle N', w', \boldsymbol{s} +\!\!+ \boldsymbol{s'}\rangle$. Intuitively, $\boldsymbol{s}$ contains a maximal number of samples to reduce $M[\boldsymbol{r}/\boldsymbol{\theta}]$. Let $\mathbb{T}_{\max}$ be the set of maximal $(\boldsymbol{r}, \boldsymbol{s})$.

Note that $\mathbb{T}_{\mathrm{term}} \subseteq \mathbb{T}_{\max}$ and there are terms for which the inclusion is strict (e.g. for the diverging term $M \equiv \mathbf{Y}(\lambda f.\, f)$, $[] \in \mathbb{T}_{\max}$ but $[] \notin \mathbb{T}_{\mathrm{term}}$). Besides, $\mathbb{T}_{\max}$ is measurable because, thanks to Proposition 3.14, for every $n \in \mathbb{N}$,

$$\{(\boldsymbol{r}, \boldsymbol{s}) \in \mathbb{R}^m \times \mathbb{S}_n \mid \langle M[\underline{\boldsymbol{r}}/\boldsymbol{\theta}], 1, []\rangle \to^* \langle N, w, \boldsymbol{s}\rangle\} = \bigcup_{\substack{U : \langle\!\langle M, \mathit{1}, \mathbb{R}^m \rangle\!\rangle \Rightarrow^* \\ \langle\!\langle \mathcal{N}, w, U \rangle\!\rangle}} U \cap (\mathbb{R}^m \times \mathbb{S}_n)$$

and the RHS is a countable union of measurable sets (Lemmas 3.12 and 3.13).

The following is a consequence of the definition of almost sure termination and a corollary of Fubini's theorem:

Lemma 3.17. *If M terminates almost surely then $\mu(\mathbb{T}_{\max} \setminus \mathbb{T}_{\mathrm{term}}) = 0$.*

Proof. Let $T \in \mathcal{B}^m$ be such that $\mu(\mathbb{R}^m \setminus T) = 0$ and for every $\boldsymbol{r} \in T$, $M[\underline{\boldsymbol{r}}/\boldsymbol{\theta}]$ terminates almost surely. For $\boldsymbol{r} \in \mathbb{R}^m$ we use the abbreviations

$$\mathbb{T}_{\boldsymbol{r},\max} \coloneqq \{\boldsymbol{s} \in \mathbb{S} \mid (\boldsymbol{r}, \boldsymbol{s}) \in \mathbb{T}_{\max}\} \qquad \mathbb{T}_{\boldsymbol{r},\mathrm{term}} \coloneqq \{\boldsymbol{s} \in \mathbb{S} \mid (\boldsymbol{r}, \boldsymbol{s}) \in \mathbb{T}_{\mathrm{term}}\}$$

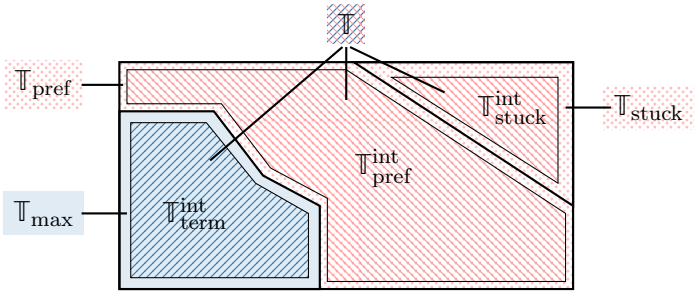


Figure 3.3: Illustration of how $\mathbb{R}^m \times \mathbb{S}$ – visualised as the entire rectangle – is partitioned to prove Theorem 3.18. The value function returns $\bot$ in the red dotted area and a closed value elsewhere (i.e. in the blue shaded area).

and we can argue analogously to $\mathbb{T}_{\text{max}}$ and $\mathbb{T}_{\text{term}}$ that they are measurable. Similarly to Lemma 3.3, for all $\boldsymbol{r} \in \mathbb{R}^m$, $\mu(\mathbb{T}_{\boldsymbol{r},\text{max}}) \leq 1$ because $(\boldsymbol{s} +\!\!+ \boldsymbol{s'}) \in \mathbb{T}_{\boldsymbol{r},\text{max}}$ and $\boldsymbol{s'} \neq []$ implies $\boldsymbol{s} \notin \mathbb{T}_{\boldsymbol{r},\text{max}}$.

Therefore, for every $\boldsymbol{r} \in T$, $\mu(\mathbb{T}_{\boldsymbol{r},\text{max}} \setminus \mathbb{T}_{\boldsymbol{r},\text{term}}) = 0$. Finally, due to a consequence of Fubini's theorem (Lemma A.4) and the fact that the Lebesgue measure is σ-finite,

$$\mu(\mathbb{T}_{\text{max}} \setminus \mathbb{T}_{\text{term}}) = \mu(\{(\boldsymbol{r}, \boldsymbol{s}) \in \mathbb{R}^m \times \mathbb{S} \mid \boldsymbol{s} \in \mathbb{T}_{\boldsymbol{r},\text{max}} \setminus \mathbb{T}_{\boldsymbol{r},\text{term}}) = 0 \qquad \square$$

Now, observe that for all $(\boldsymbol{r}, \boldsymbol{s}) \in \mathbb{R}^m \times \mathbb{S}$, exactly one of the following holds:

(i) $(\boldsymbol{r}, \boldsymbol{s})$ is maximal
(ii) for a proper prefix $\boldsymbol{s'}$ of $\boldsymbol{s}$, $(\boldsymbol{r}, \boldsymbol{s'})$ is maximal
(iii) $(\boldsymbol{r}, \boldsymbol{s})$ is *stuck*, because $\boldsymbol{s}$ does not contain enough randomness.

Formally, we say $(\boldsymbol{r}, \boldsymbol{s})$ is *stuck* if $\langle M[\underline{\boldsymbol{r}}/\boldsymbol{\theta}], 1, []\rangle \to^* \langle E[\mathbf{sample}], w, \boldsymbol{s}\rangle$, and we let $\mathbb{T}_{\text{stuck}}$ be the set of all $(\boldsymbol{r}, \boldsymbol{s})$ which get stuck. Thus,

$$\mathbb{R}^m \times \mathbb{S} = \mathbb{T}_{\text{max}} \cup \mathbb{T}_{\text{pref}} \cup \mathbb{T}_{\text{stuck}}$$

where $\mathbb{T}_{\text{pref}} := \{(\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) \mid (\boldsymbol{r}, \boldsymbol{s}) \in \mathbb{T}_{\text{max}} \wedge \boldsymbol{s'} \neq []\}$, and the union is disjoint.

Defining $\mathbb{T}^{\text{int}}_{\text{stuck}} := \bigcup\{\mathring{U} \mid \langle\!\langle M, 1, \mathbb{R}^m\rangle\!\rangle \Rightarrow^* \langle\!\langle \mathcal{E}[\mathbf{sample}], w, U\rangle\!\rangle\}$ we can argue analogously to Eq. (3.2) that $\mu(\mathbb{T}_{\text{stuck}} \setminus \mathbb{T}^{\text{int}}_{\text{stuck}}) = 0$.

Moreover, for $\mathbb{T}^{\text{int}}_{\text{pref}} := \{(\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) \mid (\boldsymbol{r}, \boldsymbol{s}) \in \mathbb{T}^{\text{int}}_{\text{term}} \text{ and } [] \neq \boldsymbol{s'} \in \mathbb{S}\}$ it holds

$$\mathbb{T}_{\text{pref}} \setminus \mathbb{T}^{\text{int}}_{\text{pref}} = \bigcup_{n \in \mathbb{N}} \left\{(\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) \mid (\boldsymbol{r}, \boldsymbol{s}) \in \mathbb{T}_{\text{max}} \setminus \mathbb{T}^{\text{int}}_{\text{term}} \wedge \boldsymbol{s'} \in \mathbb{S}_n\right\}$$

and hence, $\mu(\mathbb{T}_{\text{pref}} \setminus \mathbb{T}^{\text{int}}_{\text{pref}}) \leq \sum_{n \in \mathbb{N}} \mu(\mathbb{T}_{\text{max}} \setminus \mathbb{T}^{\text{int}}_{\text{term}}) \leq 0$.

Finally, we define

$$\mathbb{T} := \mathbb{T}^{\text{int}}_{\text{term}} \cup \mathbb{T}^{\text{int}}_{\text{pref}} \cup \mathbb{T}^{\text{int}}_{\text{stuck}}$$

Clearly, this is an open set and the situation is illustrated in Fig. 3.3. By what we have seen,

$$\mu\left((\mathbb{R}^m \times \mathbb{S}) \setminus \mathbb{T}\right) = \mu(\mathbb{T}_{\text{term}} \setminus \mathbb{T}^{\text{int}}_{\text{term}}) + \mu(\mathbb{T}_{\text{pref}} \setminus \mathbb{T}^{\text{int}}_{\text{pref}}) + \mu(\mathbb{T}_{\text{stuck}} \setminus \mathbb{T}^{\text{int}}_{\text{stuck}}) = 0$$

Moreover, to conclude the proof of our main result Theorem 3.18 it suffices to note:

(i) weight_M and value_M are differentiable everywhere on $\mathbb{T}^{\mathrm{int}}_{\mathrm{term}}$ (as for Theorem 3.16), and
(ii) $\mathrm{weight}_M(\boldsymbol{r}, \boldsymbol{s}) = 0$ and $\mathrm{value}_M(\boldsymbol{r}, \boldsymbol{s}) = \bot$ for $(\boldsymbol{r}, \boldsymbol{s}) \in \mathbb{T}^{\mathrm{int}}_{\mathrm{pref}} \cup \mathbb{T}^{\mathrm{int}}_{\mathrm{stuck}}$.

Theorem 3.18. *Let M be an SPCF term (possibly with free variables of type R) which terminates almost surely. Then its weight function weight_M and value function value_M are differentiable almost everywhere.*

We remark that almost sure termination was not used in our development until the proof of Lemma 3.17. For Theorem 3.18 we could have instead directly assumed the conclusion of Lemma 3.17; that is, almost all maximal traces are terminating. This is a strictly weaker condition than almost sure termination. The exposition we give is more appropriate: almost sure termination is a standard notion, and the development of methods to prove almost sure termination is a subject of active research.

We also note that the technique used in this chapter to establish almost everywhere differentiability could be used to target another "almost everywhere" property instead: one can simply remove the requirement that elements of Op are differentiable, and replace it with the desired property. A basic example of this is *smoothness*.

3.4 Conclusion and Related Work

We have solved an open problem in the theory of probabilistic programming: under mild assumptions densities of almost surely terminating programs are differentiable almost everywhere. This is mathematically interesting, and motivated the development of stochastic symbolic execution, a more informative form of operational semantics in this context. The result is also of major practical interest, since almost everywhere differentiability is necessary for correct gradient-based inference.

Related Work

The problem was partially addressed in the work of [Zhou et al., 2019] who prove a restricted form of our theorem for recursion-free first-order programs with analytic primitives. Our stochastic symbolic execution is related to their *compilation scheme*, which we extend to a more general language.

The idea of considering the possible control paths through a probabilistic program is fairly natural and not new to this paper; it has been used towards the design of specialised inference algorithms for probabilistic programming, see [Chaganty et al., 2013, Zhou et al., 2020]. To our knowledge, this is the first semantic formalisation of the concept, and the first time it is used to reason about whole-program density.

[Mazza and Pagani, 2021] study the correctness of automatic differentiation (AD) of purely *deterministic* programs. This problem is orthogonal to the work reported here, but it is interesting to combine their result with ours. Specifically, we show a.e. differentiability whilst [Mazza and Pagani, 2021] proves a.s. correctness of AD on the *differentiable* domain. Combining both results one concludes that for a deterministic program, AD returns a correct gradient a.s. on the *entire* domain. Going deeper into the comparison, Mazza and Pagani propose a notion of admissible primitive function strikingly similar to ours: given continuity, their condition 2 and our condition 3.1(iii) are equivalent. On the other hand we require admissible functions to be differentiable, when they are merely

continuous in [Mazza and Pagani, 2021]. Finally, we conjecture that "stable points", a central notion in [Mazza and Pagani, 2021], have a clear counterpart within our framework: for a symbolic evaluation path arriving at $\langle\!\langle \mathcal{V}, w, U \rangle\!\rangle$, for $\mathcal{V}$ a symbolic value, the points of $\mathring{U}$ are precisely the stable points.

Our work is also connected to recent developments in differentiable programming. Lee et al. [Lee et al., 2020a] study the family of *piecewise functions under analytic partition*, or just "PAP" functions. PAP functions are a well-behaved family of almost everywhere differentiable functions, which can be used to reason about automatic differentiation in recursion-free first-order programs. Since publication of our paper [Mak et al., 2021], [Huot et al., 2023] have extended the approach and given a denotational semantics based on the new category of ωPAP spaces. As an application, they recover Theorem 3.16[7].

[7]It is not clear whether they can also prove the Main Theorem 3.18 of Chapter 3, which requires reasoning about traces *not* resulting in values.

Chapter 4

A PL Framework for Stochastic Optimisation

In this chapter we present a programming language framework for stochastic optimisation via the reparameterisation gradient estimator. We assume that distributions are reparameterised (either by the user or automatically [Gorinova et al., 2020]). Thus, in the light of Eq. (2.4), the Optimisation Problem 2.10 is

$$\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[f(\boldsymbol{\theta}, \boldsymbol{\varphi_\theta}(\boldsymbol{s}))]$$

where each $\boldsymbol{\varphi_\theta} : \mathbb{R}^n \to \mathbb{R}^n$ is a diffeomorphism[1].

We generalise the problem by allowing the user to specify both the probability distribution $\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}$ and the function $(\boldsymbol{\theta}, \boldsymbol{s}) \mapsto f(\boldsymbol{\theta}, \boldsymbol{\varphi_\theta}(\boldsymbol{s}))$ *within* a programming language. For instance, for the program

$$(\lambda z.\, \mathbf{if}\ z < 0\ \mathbf{then}\ 0\ \mathbf{else}\ z)\,(\varphi_{\mu,\sigma}(\mathbf{sample}\,\mathcal{N}))$$

where $\varphi_{\mu,\sigma}(s) \coloneqq s \cdot \sigma + \mu$, this amounts to optimising[2]

$$\mathbb{E}_{s\sim\mathcal{N}(0,1)}[\mathrm{ReLU}(\varphi_{\mu,\sigma}(s))] = \mathbb{E}_{z\sim\mathcal{N}(\mu,\sigma^2)}[\mathrm{ReLU}(z)]$$

as a function of $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}_{>0}$.

As our basis we take a modified fragment of the language of Section 2.2 without recursion and scoring which makes reparameterisations explicit.

To simplify the presentation and proofs, we will pose an optimisation problem using an *eager*[3] semantics, which requires samples for all branches (no matter which branch ends up being taken). This ensures that samples are drawn independently from simple base distributions and transformed by a single diffeomorphism globally across all branches, which will prove technically more convenient than optimising the expectation of the value function w.r.t. the weight-density. Intuitively, eagerness does not affect the objective function because the "unused randomness" can be simply marginalised out.

[1] This will prove technically convenient; see e.g. Eq. (4.1) below.

[2] ReLU is the *rectified linear unit* function, which is the identity for non-negative arguments and constantly 0 otherwise.

[3] Our use of the term "eager" is unusual but taking samples for both branches is even more eager than a typical call-by-value evaluation strategy, which just evaluates the relevant branch (and thus only needs the corresponding samples).

Example 4.1. For the term

$$M \equiv (\lambda z_1.\, \mathbf{if}\ z_1 < 0\ \mathbf{then}\ (\mathbf{sample}_{\mathcal{E}} + \mathbf{sample}_{\mathcal{E}})\ \mathbf{else}\ (\varphi_{\mu,\sigma}(\mathbf{sample}_{\mathcal{N}})))\ (\varphi_{\mu,\sigma}(\mathbf{sample}_{\mathcal{N}}))$$

where $\mathcal{E}$ is the exponential distribution with parameter 1, the weight- and value-function are complex:

$$\mathrm{weight}_M((\mu,\sigma),\boldsymbol{s}) = \begin{cases} \mathcal{N}(s_1)\cdot\mathcal{E}(s_2)\cdot\mathcal{E}(s_3) & \text{if } \boldsymbol{s}=[s_1,s_2,s_3] \text{ and } \varphi_{\mu,\sigma}(s_1)<0 \\ \mathcal{N}(s_1)\cdot\mathcal{N}(s_2) & \text{if } \boldsymbol{s}=[s_1,s_2] \text{ and } \varphi_{\mu,\sigma}(s_1)\geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{value}_M((\mu,\sigma),\boldsymbol{s}) = \begin{cases} \underline{s_2+s_3} & \text{if } \boldsymbol{s}=[s_1,s_2,s_3] \text{ and } \varphi_{\mu,\sigma}(s_1)<0 \\ \underline{\varphi_{\mu,\sigma}(s_2)} & \text{if } \boldsymbol{s}=[s_1,s_2] \text{ and } \varphi_{\mu,\sigma}(s_1)\geq 0 \\ \bot & \text{otherwise} \end{cases}$$

On the other hand, the eager interpretation uses the *fixed* trace length[4] 4

$$[\![M]\!]((\mu,\sigma),[s_1,s_2,s_3,s_4]) = \begin{cases} s_2+s_3 & \text{if } \varphi_{\mu,\sigma}(s_1)<0 \\ \varphi_{\mu,\sigma}(s_4) & \text{otherwise} \end{cases}$$

and the "unnecessary randomness" can be integrated out:

$$\begin{aligned} &\mathbb{E}_{s_1\sim\mathcal{N},s_2\sim\mathcal{E},s_3\sim\mathcal{E},s_4\sim\mathcal{N}}[[\![M]\!]([s_1,\ldots,s_4])] \\ &= \mathbb{E}_{s_1\sim\mathcal{N},s_2\sim\mathcal{E},s_3\sim\mathcal{E},s_4\sim\mathcal{N}}[[\varphi_{\mu,\sigma}(s_1)<0]\cdot(s_2+s_3)] \\ &\quad + \mathbb{E}_{s_1\sim\mathcal{N},s_2\sim\mathcal{E},s_3\sim\mathcal{E},s_4\sim\mathcal{N}}[[\varphi_{\mu,\sigma}(s_1)\geq 0]\cdot\varphi_{\mu,\sigma}(s_4)] \\ &= \mathbb{E}_{s_1\sim\mathcal{N},s_2\sim\mathcal{E},s_3\sim\mathcal{E}}[[\varphi_{\mu,\sigma}(s_1)<0]\cdot(s_2+s_3)] + \mathbb{E}_{s_1\sim\mathcal{N},s_4\sim\mathcal{N}}[[\varphi_{\mu,\sigma}(s_1)\geq 0]\cdot\varphi_{\mu,\sigma}(s_4)] \\ &= \mathbb{E}_{s_1\sim\mathcal{N},s_2\sim\mathcal{E},s_3\sim\mathcal{E}}[[\varphi_{\mu,\sigma}(s_1)<0]\cdot(s_2+s_3)] + \mathbb{E}_{s_1\sim\mathcal{N},\,s_2\sim\mathcal{N}}[[\varphi_{\mu,\sigma}(s_1)\geq 0]\cdot\varphi_{\mu,\sigma}(s_2)] \\ &= \int \mathrm{weight}_M(\boldsymbol{s})\cdot\mathrm{result}_M(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s} \end{aligned}$$

where $\mathrm{result}_M(\boldsymbol{s}) = r$ provided $\mathrm{value}_M(\boldsymbol{s}) = \underline{r}$.

Eagerness will also be crucial in the next chapter for our smoothed semantics, which is the basis of our approach for addressing the bias of the reparameterisation estimator and effectively interprets conditionals by convex combinations of its branches.

Outline of the chapter. We proceed to introducing the syntax and a type system for the modified language in Section 4.1. Based on a denotational semantics (presented in Section 4.2), we formally state (in Section 4.3) the stochastic optimisation problem we are solving in the remainder of this thesis, and Section 4.4 presents a simplified framework. In Section 4.5, we guarantee that the optimisation problem is well-defined with suitable assumptions and a type system. We present an abstraction of the latter in Section 4.6, which we instantiate throughout the remainder of the thesis. Finally, we conclude and present related work in Section 4.7.

[4] s_2 and s_3 are used in the evaluation of the if-branch, whereas s_4 is used in the evaluation of the else-branch.

4.1 Syntax and Trace-Based Type System

To formalise the stochastic optimisation problem we propose a type system based on trace types, which precisely characterises for a program its space of possible eager execution traces [Lew et al., 2020]. The technical advantage is that for every (closed) term-in-context, $\mid [\mathcal{D}_1, \ldots, \mathcal{D}_n] \vdash_{\mathrm{tr}} M : \tau$, M reduces to a (unique) value using exactly the traces of the length encoded in the typing, i.e., n. So in this sense, the semantics is "total". There is no recursion and the typing ensures that there is no partiality caused by partial primitives such as $1/x$.

Types

Types are generated from *base types* (R and $R_{>0}$, the reals and positive reals) and *trace types* (typically Σ, which is a finite list of probability distributions) as well as by a *trace-based function space* constructor of the form $\tau \bullet \Sigma \to \tau'$. Formally, types are defined by the following grammar:

trace types	$\Sigma ::= [\mathcal{D}_1, \ldots, \mathcal{D}_n]$	$n \geq 0$
base types	$\iota ::= R \mid R_{>0}$	
types	$\tau ::= \iota \mid \tau \bullet \Sigma \to \tau$	

where $\mathcal{D}_i$ are probability distributions. Intuitively, a trace type is a description of the space of execution traces of a probabilistic program. The subtyping of types, as defined in Fig. 4.1a, is essentially standard.

We use list concatenation notation $+\!\!+$ for trace types, and the shorthand $\tau_1 \to \tau_2$ for function types of the form $\tau_1 \bullet [] \to \tau_2$. Intuitively, a term has type $\tau \bullet \Sigma \to \tau'$ if, when given a value of type τ, it reduces to a value of type τ' using all the samples in Σ.

Raw Terms

Henceforth, we fix dedicated variables $\theta_1 : \iota_1, \ldots, \theta_m : \iota_m$, which we refer to as ***parameters***, and a ***parameter space*** $\boldsymbol{\Theta} \subseteq [\![\iota_1]\!] \times \cdots \times [\![\iota_m]\!] \subseteq \mathbb{R}^m$, where $[\![R]\!] = \mathbb{R}$ and $[\![R_{>0}]\!] = \mathbb{R}_{>0}$.

To make reparameterisations more explicit, we use the transformed sample construct $\mathbf{sample}_{\mathcal{D}} \rhd \varphi_{\boldsymbol{\theta}}$, which can be thought of as syntactic sugar for $\underline{\varphi_{\boldsymbol{\theta}}}(\mathbf{sample}_{\mathcal{D}})$.

Definition 4.2. Let $U \subseteq \mathbb{R}^n$ be open.

(i) A differentiable function $\boldsymbol{\varphi} : U \to U$ is a ***diffeomorphism*** if it is bijective and has a differentiable inverse. (In particular, $|\det \mathbf{J}\boldsymbol{\varphi}(\boldsymbol{s})| > 0$ for all $\boldsymbol{s} \in U$.)

(ii) A function $\boldsymbol{\varphi}_{(-)} : \boldsymbol{\Theta} \times U \to U$ is a ***strong diffeomorphism*** if $\boldsymbol{\varphi}_{\boldsymbol{\theta}} : U \to U$ is a diffeomorphism for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and $\inf_{(\boldsymbol{\theta},\boldsymbol{s}) \in \boldsymbol{\Theta} \times U} |\det \mathbf{J}\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s})| > 0$.

Example 4.3. Suppose $\boldsymbol{\Theta} \subseteq \mathbb{R}^m$ is compact, $f : \boldsymbol{\Theta} \to \mathbb{R}_{>0}$ is continuous and $g : \boldsymbol{\Theta} \to \mathbb{R}$. Then the location-scale transformation

$$\varphi_{\boldsymbol{\theta}}(s) \coloneqq f(\boldsymbol{\theta}) \cdot s + g(\boldsymbol{\theta})$$

is a strong diffeomorphism because $\inf_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} f(\boldsymbol{\theta}) > 0$ ($\boldsymbol{\Theta}$ is compact and f is continuous).

$$\frac{}{\iota \sqsubseteq \iota} \qquad \frac{}{R_{>0} \sqsubseteq R} \qquad \frac{\tau_1' \sqsubseteq \tau_1 \quad \tau_2 \sqsubseteq \tau_2'}{(\tau_1 \bullet \Sigma \to \tau_2) \sqsubseteq (\tau_1' \bullet \Sigma \to \tau_2')}$$

(a) Subtyping

$$\frac{\Gamma \mid \Sigma \vdash_{\mathrm{tr}} M : \tau}{\Gamma \mid \Sigma \vdash_{\mathrm{tr}} M : \tau'}\ \tau \sqsubseteq \tau' \qquad \frac{}{x_1 : \tau_1, \ldots, x_m : \tau_m \mid [] \vdash_{\mathrm{tr}} x_i : \tau_i}$$

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\mathrm{tr}} M_1 : \iota_1 \quad \cdots \quad \Gamma \mid \Sigma_\ell \vdash_{\mathrm{tr}} M_\ell}{\Gamma \mid \Sigma_1 +\!\!+ \cdots +\!\!+ \Sigma_\ell \vdash_{\mathrm{tr}} \underline{f}(M_1, \ldots, M_\ell) : \iota}\ f : [\![\iota_1]\!] \times \cdots [\![\iota_\ell]\!] \to [\![\iota]\!] \in \mathrm{Op}$$

$$\frac{\Gamma \mid \Sigma \vdash_{\mathrm{tr}} L : R \quad \Gamma \mid \Sigma' \vdash_{\mathrm{tr}} M : \iota \quad \Gamma \mid \Sigma'' \vdash_{\mathrm{tr}} N : \iota}{\Gamma \mid \Sigma +\!\!+ \Sigma' +\!\!+ \Sigma'' \vdash_{\mathrm{tr}} \mathbf{if}\, L < 0\, \mathbf{then}\, M\, \mathbf{else}\, N : \iota}$$

$$\frac{\Gamma, y : \tau_1 \mid \Sigma \vdash_{\mathrm{tr}} M : \tau_2}{\Gamma \mid [] \vdash_{\mathrm{tr}} \lambda y.\, M : \tau_1 \bullet \Sigma \to \tau_2} \qquad \frac{\Gamma \mid \Sigma_1 \vdash_{\mathrm{tr}} M : \tau_1 \bullet \Sigma_3 \to \tau_2 \quad \Gamma \mid \Sigma_2 \vdash_{\mathrm{tr}} N : \tau_1}{\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 +\!\!+ \Sigma_3 \vdash_{\mathrm{tr}} M\, N : \tau_2}$$

$$\frac{}{\boldsymbol{\theta}, \Gamma \mid [\mathcal{D}] \vdash_{\mathrm{tr}} \mathbf{sample}_{\mathcal{D}} \rhd \varphi_{\boldsymbol{\theta}} : R}\ \varphi_{(-)} \in \mathrm{Diff}_{\boldsymbol{\Theta}}(\mathrm{supp}(\mathcal{D}))$$

(b) Typing judgments

Figure 4.1: A basic trace-based type system

Assumption 4.4. For open $U \subseteq \mathbb{R}$, $\mathrm{Diff}_{\boldsymbol{\Theta}}(U)$ is a set of polynomials $\varphi_{(-)} : \boldsymbol{\Theta} \times \mathbb{R} \to \mathbb{R}$ which are strong diffeomorphisms.

The *raw terms* of the modified programming language are given by

$$\begin{aligned} M ::= {} & x \mid \underline{f}(M, \ldots, M) \mid \mathbf{if}\, M < 0\, \mathbf{then}\, M\, \mathbf{else}\, M \\ & \mid \lambda x.\, M \mid M\, M \\ & \mid \mathbf{sample}_{\mathcal{D}} \rhd \varphi_{\boldsymbol{\theta}} \end{aligned}$$

where $f \in \mathrm{Op}$, $\mathcal{D} \in \mathrm{Dist}$ and $\varphi_{(-)} \in \mathrm{Diff}_{\boldsymbol{\Theta}}(\mathrm{supp}(\mathcal{D}))$.

We can express affine transformations, in particular, the location-scale transformations as follows:

Example 4.5 (Location-Scale Transformation)**.** As expressed in Eq. (2.3), we can obtain samples from $\mathcal{N}(\mu, \sigma^2)$ via $\mathbf{sample}_{\mathcal{N}} \rhd \varphi_{\mu,\sigma}$, where $\varphi_{\mu,\sigma}(s) := s \cdot \sigma + \mu$, which is syntactic sugar for the term $\mathbf{sample}_{\mathcal{N}} \cdot \sigma + \mu$.

In order to comply with Assumption 4.4, we require $(\mu, 0) \notin \boldsymbol{\Theta}$ because $\varphi_{\mu,0}$ is *not* bijective.

Typing Judgements

Dual context *typing judgements* of the form, $\Gamma \mid \Sigma \vdash_{\mathrm{tr}} M : \tau$, are defined in Fig. 4.1b, where $\Gamma = x_1 : \tau_1, \cdots, x_m : \tau_m$ is a finite map describing a set of variable-type bindings; and the trace type Σ precisely captures the distributions from which samples are drawn in a (fully eager) call-by-value evaluation of the term M.

In particular the following rules are admissible (provided the corresponding primitives are in Op):

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\mathrm{tr}} M : R \quad \Gamma \mid \Sigma_2 \vdash_{\mathrm{tr}} N : R}{\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 \vdash_{\mathrm{tr}} M \circ N : R} \circ \in \{+, \cdot\}$$

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\mathrm{tr}} M : R_{>0} \quad \Gamma \mid \Sigma_2 \vdash_{\mathrm{tr}} N : R_{>0}}{\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 \vdash_{\mathrm{tr}} M \circ N : R_{>0}} \circ \in \{+, \cdot\}$$

$$\frac{\Gamma \mid \Sigma \vdash_{\mathrm{tr}} M : R}{\Gamma \mid \Sigma \vdash_{\mathrm{tr}} \underline{\exp}(M) : R_{>0}} \qquad \frac{\Gamma \mid \Sigma \vdash_{\mathrm{tr}} M : R_{>0}}{\Gamma \mid \Sigma \vdash_{\mathrm{tr}} \underline{\log}(M) : R}$$

Example 4.6 (Encoding the ELBO for Variational Inference)**.** We consider the example used by [Lee et al., 2018] to prove the biasedness of the reparameterisation gradient (Fig. 2.4). The density is

$$p(z) \coloneqq \mathcal{N}(z \mid 0, 1) \cdot \begin{cases} \mathcal{N}(0 \mid -2, 1) & \text{if } z < 0 \\ \mathcal{N}(0 \mid 5, 1) & \text{otherwise} \end{cases}$$

and they use a variational family with density $q_\theta(z) \coloneqq \mathcal{N}(z \mid \theta, 1)$, which is reparameterised using a standard normal noise distribution and transformation $\varphi(\theta, s) \coloneqq s + \theta$.

First, we define an auxiliary term for the pdf of normals with mean m and standard derivation s:

$$N \equiv \lambda x, m, s.\, \left(\sqrt{2\pi} \cdot s\right)^{-1} \cdot \exp\left(-0.5 \cdot \left((x - m) \cdot s^{-1}\right)^2\right)$$

(We assume that Op contains the constants $\sqrt{2\pi}$ as well as the standard arithmetic functions.) Then, we can define

$$M \equiv \big(\lambda z.\, \underbrace{\log\left(N\, z\, 0\, 1\right) + \left(\mathbf{if}\, z < 0\, \mathbf{then}\, \log\left(N\, 0\, (-2)\, 1\right)\, \mathbf{else}\, \log\left(N\, 0\, 5\, 1\right)\right)}_{\log p} - \underbrace{\log\left(N\, z\, \theta\, 1\right)}_{\log q_\theta}\big)\, \left(\mathbf{sample}_{\mathcal{N}} \rhd \varphi_\theta\right)$$

Note that $\mid [] \vdash_{\mathrm{tr}} N : R \to R \to R_{>0} \to R_{>0}$ and $\theta : R \mid [\mathcal{N}] \vdash_{\mathrm{tr}} M : R$.

Conditionals

We assume that the branches of conditionals have a base type. Otherwise it would not be clear how to type terms such as[5]

$$M \equiv \mathbf{if}\, x < 0\, \mathbf{then}\, (\lambda x.\, \mathbf{sample}_{\mathcal{N}})\, \mathbf{else}\, (\lambda x.\, \mathbf{sample}_{\mathcal{E}} + \mathbf{sample}_{\mathcal{E}})$$
$$N \equiv (\lambda f.\, f\, (f\, \mathbf{sample}_{\mathcal{N}}))\, M$$

because the branches draw a different number of samples from different distributions, and have types $R \bullet [\mathcal{N}] \to R$ and $R \bullet [\mathcal{E}, \mathcal{E}] \to R$, respectively. However, for

$$M' \equiv \mathbf{if}\, x < 0\, \mathbf{then}\, \mathbf{sample}_{\mathcal{N}}\, \mathbf{else}\, \mathbf{sample}_{\mathcal{E}} + \mathbf{sample}_{\mathcal{E}}$$

[5] Since the transformation does not matter here, we omit it and simply write $\mathbf{sample}_{\mathcal{N}}$ instead of $\mathbf{sample}_{\mathcal{N}} \rhd \mathrm{id}$, etc.

we can (safely) type

$$\begin{aligned} x : R \mid [\mathcal{N}, \mathcal{E}, \mathcal{E}] &\vdash_{\mathrm{tr}} M' : R \\ \mid [] &\vdash_{\mathrm{tr}} \lambda x.\, M' : R \bullet [\mathcal{N}, \mathcal{E}, \mathcal{E}] \to R \\ \mid [\mathcal{N}, \mathcal{N}, \mathcal{E}, \mathcal{E}, \mathcal{N}, \mathcal{E}, \mathcal{E}] &\vdash_{\mathrm{tr}} (\lambda f.\, f\,(f\,\mathbf{sample}_{\mathcal{N}}))\,(\lambda x.\, M') : R \end{aligned}$$

Example 4.7. Consider the following terms:

$$\begin{aligned} L &\equiv \lambda x.\, \mathbf{sample}_{\mathcal{N}} + \mathbf{sample}_{\mathcal{N}} \\ M &\equiv \mathbf{if}\, x < 0\, \mathbf{then}\, (\lambda y.\, y + y)\, \mathbf{sample}_{\mathcal{N}}\, \mathbf{else}\, (\mathbf{sample}_{\mathcal{N}} + \mathbf{sample}_{\mathcal{N}}) \end{aligned}$$

We can derive the following typing judgements:

$$\begin{aligned} \mid [] &\vdash_{\mathrm{tr}} L : R_{>0} \bullet [\mathcal{N}, \mathcal{N}] \to R \\ x : R_{>0} \mid [\mathcal{N}, \mathcal{N}, \mathcal{N}] &\vdash_{\mathrm{tr}} M : R \\ \mid [] &\vdash_{\mathrm{tr}} \lambda x.\, M : R_{>0} \bullet [\mathcal{N}, \mathcal{N}, \mathcal{N}] \to R \\ \mid [\mathcal{N}, \mathcal{N}, \mathcal{N}, \mathcal{N}] &\vdash_{\mathrm{tr}} (\lambda x.\, M)\, \mathbf{sample}_{\mathcal{N}} : R \\ \mid [\mathcal{N}, \mathcal{N}] &\vdash_{\mathrm{tr}} (\lambda f.\, f\,(f\,0))\,(\lambda x.\, \mathbf{sample}_{\mathcal{N}}) : R \end{aligned}$$

Note that $\mathbf{if}\, x < 0\, \mathbf{then}\, (\lambda x.\, \mathbf{sample}_{\mathcal{N}})\, \mathbf{else}\, (\lambda x.\, x)$ is *not* typable.

Finally, trace types are unique:

Lemma 4.8. *If* $\Gamma \mid \Sigma \vdash_{\mathrm{tr}} M : \tau$ *and* $\Gamma \mid \Sigma' \vdash_{\mathrm{tr}} M : \tau'$ *then* $\Sigma = \Sigma'$.

Proof sketch. We define an equivalence relation $\approx$ on types by

(i) $\iota \approx \iota'$

(ii) $(\tau_1 \bullet \Sigma \to \tau_2) \approx (\tau_1' \bullet \Sigma' \to \tau_2')$ iff $\tau_1 \approx \tau_1'$ implies $\Sigma = \Sigma'$ and $\tau_2 \approx \tau_2'$

Intuitively, two types are related by $\approx$ if for (inductively) related arguments they draw the same samples and again have related return types. We extend the relation to contexts: $\Gamma \approx \Gamma'$ if for all $x : \tau$ in Γ and $x : \tau'$ in Γ', $\tau \approx \tau'$.

Then we show by induction that if $\Gamma \mid \Sigma \vdash_{\mathrm{tr}} M : \tau$, $\Gamma' \mid \Sigma' \vdash_{\mathrm{tr}} M : \tau'$ and $\Gamma \approx \Gamma'$ then $\Sigma = \Sigma'$ and $\tau \approx \tau'$. Finally, this strengthened statement allows us to prove the tricky case of the lemma: application. □

4.2 Denotational Value Semantics

In Section 2.2.2 we presented operational semantics for our language interpreting conditionals lazily, which is especially important in the presence of recursion. In this subsection we present an eager and denotational version of this semantics, which will render reasoning about (correctness) properties easier. Since we are ultimately interested in expectations (see also Problem 4.12 below), we need to ensure measurability.

It is well-known that the category of measurable spaces and measurable functions is not cartesian closed [Aumann, 1961], which means that there is no interpretation of the lambda calculus as measurable functions. These difficulties led [Heunen et al., 2017a] to develop the category **QBS** of *quasi-Borel spaces (QBS)*.

Like measurable space (X, Σ_X), a ***quasi-Borel space (QBS)*** is a pair (X, M_X) where X is a set; but instead of axiomatising the measurable subsets Σ_X, QBS axiomatises the *admissible random elements* M_X. The set M_X, which is a collection of functions $\mathbb{R} \to X$, must satisfy the following closure properties:

- if $\alpha \in M_X$ and $f : \mathbb{R} \to \mathbb{R}$ is measurable, then $\alpha \circ f \in M_X$
- if $\alpha : \mathbb{R} \to X$ is constant then $\alpha \in M_X$
- given a countable partition of the reals $\mathbb{R} = \biguplus_{i \in \mathbb{N}} S_i$ where each S_i is Borel, and $\{\alpha_i\}_{i \in \mathbb{N}} \subseteq M_X$, the function $r \mapsto \alpha_i(r)$ where $r \in S_i$ is in M_X.

The **QBS** morphisms $(X, M_X) \to (Y, M_Y)$ are functions $f : X \to Y$ such that $f \circ \alpha \in M_Y$ whenever $\alpha \in M_X$. Notably, morphisms can be combined piecewisely, which we need for conditionals.

For $\tau \sqsubseteq \tau'$ there is an evident embedding morphism $\mathrm{emb}_{\tau,\tau'} : [\![\tau]\!] \to [\![\tau']\!]$ defined by

- $\mathrm{emb}_{\iota,\iota'}(x) = x$ if $\iota \sqsubseteq \iota'$ and $x \in [\![\iota]\!] \subseteq [\![\iota']\!]$
- $\mathrm{emb}_{\tau_1 \to \tau_2, \tau_1' \to \tau_2'}(f) \coloneqq x \in [\![\tau_1']\!] \mapsto \mathrm{emb}_{\tau_2,\tau_2'}(f(\mathrm{emb}_{\tau_1',\tau_1}(x)))$

We interpret our programming language in the category **QBS** of quasi-Borel spaces. Types are interpreted as follows:

$$\begin{aligned} [\![R]\!] &\coloneqq (\mathbb{R}, M_{\mathbb{R}}) \\ [\![R_{>0}]\!] &\coloneqq (\mathbb{R}_{>0}, M_{\mathbb{R}_{>0}}) \\ [\![[\mathcal{D}_1, \dots, \mathcal{D}_n]]\!] &\coloneqq (\mathbb{R}, M_{\mathbb{R}})^n \\ [\![\tau_1 \bullet \Sigma \to \tau_2]\!] &\coloneqq [\![\tau_1]\!] \times [\![\Sigma]\!] \Rightarrow [\![\tau_2]\!] \end{aligned}$$

where $M_{\mathbb{R}}$ is the set of measurable functions $\mathbb{R} \to \mathbb{R}$; similarly for $M_{\mathbb{R}_{>0}}$. (As for trace types, we use list notation (and list concatenation) for traces.)

With the following assumption we can interpret terms-in-context $\Gamma \mid \Sigma \vdash_{\mathrm{tr}} M : \tau$ as $[\![\Gamma \mid \Sigma \vdash_{\mathrm{tr}} M : \tau]\!] : [\![\Gamma]\!] \times [\![\Sigma]\!] \to [\![\tau]\!]$; see Fig. 4.2.

Assumption 4.9. Primitive operations $f \in \mathrm{Op}$ are measurable.

It is not difficult to see that this interpretation of terms-in-context is well-defined and total. For the conditional clause, we may assume that the trace type and the trace are presented as partitions $\Sigma_1 +\!\!+ \Sigma_2 +\!\!+ \Sigma_3$ and $\boldsymbol{s}_1 +\!\!+ \boldsymbol{s}_2 +\!\!+ \boldsymbol{s}_3$ respectively. This is justified because it follows from the judgement $\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 +\!\!+ \Sigma_3 \vdash_{\mathrm{tr}} \mathbf{if}\, L < 0\, \mathbf{then}\, M\, \mathbf{else}\, N : \tau$ that $\Gamma \mid \Sigma_1 \vdash_{\mathrm{tr}} L : R$, $\Gamma \mid \Sigma_2 \vdash_{\mathrm{tr}} M : \iota$ and $\Gamma \mid \Sigma_3 \vdash_{\mathrm{tr}} N : \iota$ are provable; and we know that each of Σ_1, Σ_2 and Σ_3 is unique, thanks to Lemma 4.8; their respective lengths then determine the partition $\boldsymbol{s}_1 +\!\!+ \boldsymbol{s}_2 +\!\!+ \boldsymbol{s}_3$. Similarly for the application clause, the components Σ_1 and Σ_2 are determined by Lemma 4.8, and Σ_3 by the type of M.

4.2.1 Transformed Semantics

We introduce a *transformed* version of this semantics, $[\![(-)]\!]^t$, such that the effect of the transformations can be factored out globally:

Proposition 4.10. *If $\boldsymbol{\theta} \mid [\mathcal{D}_1, \dots, \mathcal{D}_n] \vdash_{\mathrm{tr}} M : R$ then there exists a polynomial function $\boldsymbol{\varphi}_{(-)} : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}^n$ which is a strong diffeomorphism and for all $(\boldsymbol{\theta}, \boldsymbol{s}) \in \boldsymbol{\Theta} \times \mathbb{R}^n$,*

$$[\![M]\!](\boldsymbol{\theta}, \boldsymbol{s}) = [\![M]\!]^t(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))$$

$$
\begin{aligned}
&[\![\Gamma \mid \Sigma \vdash_{\mathrm{tr}} M : \tau']\!](\gamma, \boldsymbol{s}) \coloneqq \mathrm{emb}_{\tau,\tau'}([\![\Gamma \mid \Sigma \vdash_{\mathrm{tr}} M : \tau]\!](\gamma, \boldsymbol{s})) \\
&[\![x_1 : \tau_1, \ldots, x_m : \tau_m \mid [] \vdash_{\mathrm{tr}} x_i : \tau_i]\!](\gamma, []) \coloneqq \gamma_i \\
&[\![\Gamma \mid \Sigma_1 +\!\!+ \cdots +\!\!+ \Sigma_\ell \vdash_{\mathrm{tr}} \underline{f}(M_1, \ldots, M_\ell) : \iota]\!](\gamma, \boldsymbol{s}_1 +\!\!+ \cdots +\!\!+ \boldsymbol{s}_\ell) \coloneqq \\
&\qquad\qquad f([\![\Gamma \mid \Sigma_1 \vdash_{\mathrm{tr}} M_1 : \iota_1]\!](\gamma, \boldsymbol{s}_1), \ldots, [\![\Gamma \mid \Sigma_\ell \vdash_{\mathrm{tr}} M_\ell : \iota_\ell]\!](\gamma, \boldsymbol{s}_\ell)) \\
&[\![\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 +\!\!+ \Sigma_3 \vdash_{\mathrm{tr}} M\, N : \tau]\!](\gamma, \boldsymbol{s}_1 +\!\!+ \boldsymbol{s}_2 +\!\!+ \boldsymbol{s}_3) \coloneqq \\
&\qquad\qquad [\![\Gamma \mid \Sigma_1 \vdash_{\mathrm{tr}} M : \tau_1 \bullet \Sigma_3 \to \tau_2]\!](\gamma, \boldsymbol{s}_1)([\![\Gamma \mid \Sigma_2 \vdash_{\mathrm{tr}} N : \tau_1]\!](\gamma, \boldsymbol{s}_2), \boldsymbol{s}_3) \\
&[\![\Gamma \mid [] \vdash_{\mathrm{tr}} \lambda y.\, M : \tau_1 \bullet \Sigma \to \tau_2]\!](\gamma, []) \coloneqq \\
&\qquad\qquad (v, \boldsymbol{s}) \in [\![\tau_1]\!] \times [\![\Sigma]\!] \mapsto [\![\Gamma, x : \tau_1 \mid \Sigma \vdash_{\mathrm{tr}} M : \tau_2]\!]((\gamma, v), \boldsymbol{s}) \\
&[\![\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 +\!\!+ \Sigma_3 \vdash_{\mathrm{tr}} \mathbf{if}\, L < 0\, \mathbf{then}\, M\, \mathbf{else}\, N : \iota]\!](\gamma, \boldsymbol{s}_1 +\!\!+ \boldsymbol{s}_2 +\!\!+ \boldsymbol{s}_3)) \coloneqq \\
&\qquad\qquad \begin{cases} [\![\Gamma \mid \Sigma_2 \vdash_{\mathrm{tr}} M : \tau]\!](\gamma, \boldsymbol{s}_2) & \text{if } [\![\Gamma \mid \Sigma_1 \vdash_{\mathrm{tr}} L : R]\!](\gamma, \boldsymbol{s}_1) < 0 \\ [\![\Gamma \mid \Sigma_3 \vdash_{\mathrm{tr}} N : \tau]\!](\gamma, \boldsymbol{s}_3) & \text{otherwise} \end{cases} \\
&[\![\boldsymbol{\theta}, \Gamma \mid [\mathcal{D}] \vdash_{\mathrm{tr}} \mathbf{sample}\, \mathcal{D} \rhd \varphi_{\boldsymbol{\theta}} : R]\!]((\boldsymbol{\theta}, \gamma), [s]) \coloneqq \varphi_{\boldsymbol{\theta}}(s)
\end{aligned}
$$

Figure 4.2: Interpretation of terms-in-context $[\![\Gamma \mid \Sigma \vdash_{\mathrm{tr}} M : \tau]\!] : [\![\Gamma]\!] \times [\![\Sigma]\!] \to [\![\tau]\!]$

This decomposition will prove technically convenient later when demonstrating correctness properties.

Intuitively, the transformed semantics $[\![M]\!]^t$ assumes that the transformations have already been applied to the samples. Thus, the only rule which is modified is the one for the transformed-sample construct:

$$[\![\boldsymbol{\theta}, \Gamma \mid [\mathcal{D}] \vdash_{\mathrm{tr}} \mathbf{sample}\, \mathcal{D} \rhd \varphi_{\boldsymbol{\theta}} : R]\!]^t((\boldsymbol{\theta}, \gamma), [z]) \coloneqq z$$

Example 4.11. For the term

$$
\begin{aligned}
M \equiv (\lambda z_1.\, \mathbf{if}\, z_1 < 0\, \mathbf{then}\, (\mathbf{sample}\, \mathcal{E} + \mathbf{sample}\, \mathcal{E})\, \mathbf{else}\, (\mathbf{sample}\, \mathcal{N} \rhd \varphi_{\mu,\sigma})) \\
(\mathbf{sample}\, \mathcal{N} \rhd \varphi_{\mu,\sigma})
\end{aligned}
$$

in Example 4.1,

$$
\begin{aligned}
[\![M]\!]^t((\mu, \sigma), [z_1, z_2, z_3, z_4]) &= \begin{cases} z_2 + z_3 & \text{if } z_1 < 0 \\ z_4 & \text{otherwise} \end{cases} \\
\boldsymbol{\varphi}_{\mu,\sigma}([s_1, s_2, s_3, s_4]) &\coloneqq [\varphi_{\mu,\sigma}(s_1), s_2, s_3, \varphi_{\mu,\sigma}(s_4)]
\end{aligned}
$$

The result is intuitively obvious and will be subsumed by Proposition 5.6. In particular, for conditionals the eager semantics is crucial (cf. Example 4.1).

4.3 Problem Statement

We formally state our optimisation problem:

Problem 4.12. Optimisation

Given: term-in-context, $\theta_1 : \iota_1, \cdots, \theta_m : \iota_m \mid [\mathcal{D}_1, \ldots, \mathcal{D}_n] \vdash_{\mathrm{tr}} M : R$ and parameter space $\boldsymbol{\Theta} \subseteq [\![\iota_1]\!] \times \cdots [\![\iota_m]\!]$

Find: $\operatorname{argmin}_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathbb{E}_{s_1 \sim \mathcal{D}_1, \ldots, s_n \sim \mathcal{D}_n} [[\![M]\!](\boldsymbol{\theta}, \boldsymbol{s})]$

We will often abbreviate the product of the densities $\mathcal{D}_1, \ldots, \mathcal{D}_n$ as $\boldsymbol{\mathcal{D}}(\boldsymbol{s}) \coloneqq \prod_{i=1}^{n} \mathcal{D}_i(s_i)$. Besides, in the light of Proposition 4.10, there exists a diffeomorphism, $\boldsymbol{\varphi}_{(-)}$ such that

$$\mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}}[[\![M]\!](\boldsymbol{s})] = \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}}[[\![M]\!]^t(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))] = \mathbb{E}_{\boldsymbol{z} \sim \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}[[\![M]\!]^t(\boldsymbol{z})] \tag{4.1}$$

where $\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) \coloneqq \boldsymbol{\mathcal{D}}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) \cdot |\det \mathbf{J}\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})|$ and $\mathbf{J}$ denotes the Jacobian matrix.

4.4 Simple Function Calculus

Many of our key challenges (such as unbiasedness) are primarily caused by conditionals. To illustrate our approaches and to focus on the essentials first, we introduce a simple (1st-order) function calculus to represent discontinuous functions in a piecewise manner.

Let $z_1, \ldots, z_n$ be fixed variables (for a fixed arity n). We define a class Λ^{SFC} of (syntactic) representations of piecewisely defined functions $\mathbb{R}^n \to \mathbb{R}$ inductively:

$$\Lambda^{\mathrm{SFC}} \ni F ::= z_j \mid \underline{f}(F, \ldots, F) \mid \mathbf{if}\ F < 0\ \mathbf{then}\ F\ \mathbf{else}\ F$$

where $f : \mathbb{R}^\ell \to \mathbb{R} \in \mathrm{Op}$ is a primitive operation. As before, $F \in \Lambda^{\mathrm{SFC}}$ denotes a function $[\![F]\!] : \mathbb{R}^n \to \mathbb{R}$.

Example 4.13. Example 2.16 can be expressed as F_1:

$$\begin{aligned}
F_1' &\equiv \mathbf{if}\ z < 0\ \mathbf{then}\ 0\ \mathbf{else}\ 1 \\
F_1 &\equiv -0.5 \cdot z^2 + \mathbf{if}\ z < 0\ \mathbf{then}\ 0\ \mathbf{else}\ 1 \\
F_2 &\equiv \mathbf{if}\ (a \cdot (\mathbf{if}\ b \cdot z_1 + c < 0\ \mathbf{then}\ 0\ \mathbf{else}\ 1) \\
&\qquad + d \cdot (\mathbf{if}\ e \cdot z_2 + f < 0\ \mathbf{then}\ 0\ \mathbf{else}\ 1) + g) < 0 \\
&\quad \mathbf{then}\ 0\ \mathbf{else}\ 1
\end{aligned}$$

F_2 illustrates that nested (if-)branching can occur not only in branches but also in the guard/condition. Such nesting arises in practice (see `xornet` in Section 7.1) and facilitates writing concise models.

In particular, $F \in \Lambda^{\mathrm{SFC}}$ cannot contain parameters θ_i, to simplify issues further. Instead, we assume that the parameter-dependent transformation is provided separately:

Problem 4.14. Simple Function Calculus Optimisation

Given: $F \in \Lambda^{\mathrm{SFC}}$, the *parameter space* $\boldsymbol{\Theta} \subseteq \mathbb{R}^m$, $\mathcal{D}_1, \dots, \mathcal{D}_n \in \mathrm{Dist}$ and $\varphi^{(1)}_{(-)} \in \mathrm{Diff}_{\boldsymbol{\Theta}}(\mathrm{supp}(\mathcal{D}_1)), \dots, \varphi^{(n)}_{(-)} \in \mathrm{Diff}_{\boldsymbol{\Theta}}(\mathrm{supp}(\mathcal{D}_n))$

Find: $\mathrm{argmin}_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathbb{E}_{s_1 \sim \mathcal{D}_1, \dots, s_n \sim \mathcal{D}_n}[[\![F]\!](\varphi^{(1)}_{\boldsymbol{\theta}}(s_1), \dots, \varphi^{(n)}_{\boldsymbol{\theta}}(s_n))]$

As before, we can abbreviate the objective function as

$$\mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}}[[\![F]\!](\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))]$$

where $\boldsymbol{\mathcal{D}}(\boldsymbol{s}) \coloneqq \prod_{i=1}^{n} \mathcal{D}_i(s_i)$ and $\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \coloneqq (\varphi^{(1)}_{\boldsymbol{\theta}}(s_1), \dots, \varphi^{(n)}_{\boldsymbol{\theta}}(s_n))$. Besides, since F does not sample, $[\![F]\!] = [\![F]\!]^t$ and

$$\mathbb{E}_{\boldsymbol{z} \sim \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}[[\![F]\!](\boldsymbol{z})] = \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}}[[\![F]\!](\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))]$$

where $\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) \coloneqq \boldsymbol{\mathcal{D}}(\boldsymbol{\varphi}^{-1}_{\boldsymbol{\theta}}(\boldsymbol{z})) \cdot |\det \mathbf{J}\boldsymbol{\varphi}^{-1}_{\boldsymbol{\theta}}(\boldsymbol{z})|$.

Remark 4.15. We could have allowed $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\varphi}_{\boldsymbol{\theta}}$ which are correlated. We chose the present more restrictive setup to ensure that Problem 4.14 is a special case of Problem 4.12.

4.5 Integrability

As we have discussed in Section 2.5, the objective function in Problem 4.12 may be ill-defined due to an unsuitable choice of primitive operations and base distributions. In the light of Corollary 2.23 it appears to be sufficient to assume the following:

Assumption 4.16. The densities in Dist are (generalised) Schwartz functions (when restricted to their support, cf. Definition 2.18).

Assumption 4.17. Op is a set of measurable functions bounded by polynomials.

We will keep assuming Assumption 4.16 for the remainder of this thesis, but we will relax Assumption 4.17 in Section 4.5.1. As per Example 2.19, Dist may include e.g. (half) normal, exponential and logistic distributions.

For the simple function calculus we can prove the following via a straightforward structural induction, exploiting the closure properties of Lemma 2.25 and Corollary 2.23:

Proposition 4.18. *If $F \in \Lambda^{\mathrm{SFC}}$ then $[\![F]\!]$ is bounded by a polynomial and for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$,*

$$\mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}}[[\![F]\!](\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))] < \infty$$

Next, we would like to extend this result to terms $\boldsymbol{\theta} \mid \Sigma \vdash_{\mathrm{tr}} M : \iota$ of the full (higher-order) language. Intuitively, it is still pretty obvious that $[\![M]\!]$ is piecewisely defined and bounded by a polynomial on each piece (thus globally bounded by a polynomial). Nonetheless, we wish to prove the property formally to illustrate our proof techniques

(which extend *open logical relations* [Barthe et al., 2020a] to probabilistic terms) employed throughout the rest of this thesis. We will also relax Assumption 4.17 (whilst maintaining integrability) in Section 4.5 to include primitive operations such as exp and log required in Example 4.6 for expressing probability density functions directly.

We define a logical predicate $\mathcal{PB}^{(n)}_\tau$ on $\mathbf{\Theta} \times \mathbb{R}^n \to [\![\tau]\!]$:

(i) $f \in \mathcal{PB}^{(n)}_\iota$ if f is bounded by a polynomial
(ii) $f \in \mathcal{PB}^{(n)}_{\tau_1 \bullet \Sigma_3 \to \tau_2}$ if for all $n_2 \in \mathbb{N}$ and $g \in \mathcal{PB}^{(n+n_2)}_{\tau_1}$, $f \odot g \in \mathcal{PB}^{(n+n_2+|\Sigma_3|)}_{\tau_2}$

where for $f : \mathbb{R}^{n_1} \to [\![\tau_1 \bullet \Sigma_3 \to \tau_2]\!]$ and $g : \mathbb{R}^{n_1+n_2} \to [\![\tau_1]\!]$ we define

$$\begin{aligned} f \odot g : \mathbb{R}^{n_1+n_2+|\Sigma_3|} &\to \tau_2 \\ \boldsymbol{s}_1 +\!\!+ \boldsymbol{s}_2 +\!\!+ \boldsymbol{s}_3 &\mapsto f(\boldsymbol{s}_1)(g(\boldsymbol{s}_1 +\!\!+ \boldsymbol{s}_2), \boldsymbol{s}_3) \end{aligned}$$

Intuitively, g may depend on the samples in $\boldsymbol{s}_2$ (in addition to $\boldsymbol{s}_1$) and the function application may consume further samples $\boldsymbol{s}_3$ (as determined by the trace type Σ_3).

A straightforward induction on the definition of subtyping reveals:

Lemma 4.19. *If $f \in \mathcal{PB}^{(n)}_\tau$ and $\tau \sqsubseteq \tau'$ then $\mathrm{emb}_{\tau,\tau'}(f) \in \mathcal{PB}^{(n)}_{\tau'}$.*

Lemma 4.20 (Fundamental)**.** *If $\theta_1 : \iota_1, \dots, \theta_m : \iota_m, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma \vdash_{\mathrm{tr}} M : \tau$, $n \in \mathbb{N}$ and $\xi_1 \in \mathcal{PB}^{(n)}_{\tau_1}, \dots, \xi_\ell \in \mathcal{PB}^{(n)}_{\tau_\ell}$ then*

$$[\![M]\!]^t * \langle \xi_1, \dots, \xi_\ell \rangle \in \mathcal{PB}^{(n+|\Sigma|)}_\tau$$

where $([\![M]\!]^t * \langle \xi_1, \dots, \xi_\ell \rangle)(\boldsymbol{\theta}, \boldsymbol{s} +\!\!+ \boldsymbol{s}') \coloneqq [\![M]\!]^t(\boldsymbol{\theta}, (\xi_1(\boldsymbol{\theta}, \boldsymbol{s}), \dots, \xi_\ell(\boldsymbol{\theta}, \boldsymbol{s})), \boldsymbol{s}')$.

It is worth noting that, in contrast to more standard fundamental lemmas, here we need to capture the dependency of the free variables on some number n of further samples. E.g. in the context of $(\lambda x.\, x)\,\mathbf{sample}_{\mathcal{N}}$ the subterm x depends on a sample although this is not apparent if we consider x in isolation.

We conclude:

Proposition 4.21 (Integrability)**.** *Let $\boldsymbol{\theta} \mid \Sigma \vdash_{\mathrm{tr}} M : R$. For all $\boldsymbol{\theta} \in \mathbf{\Theta}$,*

$$\mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}}[[\![M]\!](\boldsymbol{\theta}, \boldsymbol{s})] < \infty$$

Proof. By Lemma 4.20, $[\![M]\!]^t$ is bounded by a polynomial. Hence, by Lemma 2.25(i) and Proposition 4.10, $[\![M]\!]$ is bounded by a polynomial and the claim follows with Lemma 2.20 (since all distributions in Σ have Schwartz densities). □

Proof of Lemma 4.20. We prove the claim by induction on the type derivation of M.

- For subtyping we exploit Lemma 4.19.
- For variables (incl. θ_i) this is immediate.
- Suppose $\boldsymbol{\theta}, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma_1 +\!\!+ \dots +\!\!+ \Sigma_\ell \vdash_{\mathrm{tr}} \underline{f}(M_1, \dots, M_\ell) : \iota$ because $f : [\![\iota_1]\!] \times \dots \times [\![\iota_\ell]\!] \to [\![\iota]\!] \in \mathrm{Op}$ and $\boldsymbol{\theta}, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma_i \vdash_{\mathrm{tr}} M_i : \iota_i$. Let $n \in \mathbb{N}$ and $\xi_1 \in \mathcal{PB}^{(n)}_{\tau_1}, \dots, \xi_\ell \in \mathcal{PB}^{(n)}_{\tau_\ell}$. By the inductive hypothesis,

$$[\![M_i]\!]^t * \langle \xi_1, \dots, \xi_\ell \rangle \in \mathcal{PB}^{(n+|\Sigma_i|)}_{\iota_i}$$

Let $f_i(\boldsymbol{\theta}, \boldsymbol{z} +\!\!+ \boldsymbol{z}_1 +\!\!+ \cdots +\!\!+ \boldsymbol{z}_\ell) \coloneqq (\llbracket M_i \rrbracket^t * \langle \xi_1, \ldots, \xi_\ell \rangle)(\boldsymbol{\theta}, \boldsymbol{z} +\!\!+ \boldsymbol{z}_i)$. Clearly,

$$f_i \in \mathcal{PB}_{\iota_i}^{(n+|\Sigma_1|+\cdots+|\Sigma_\ell|)}$$

By Assumption 4.17 and Lemma 2.25(i) (closure under composition),

$$f \circ \langle f_1, \ldots, f_\ell \rangle \in \mathcal{PB}_{\iota}^{(n+|\Sigma_1|+\cdots+|\Sigma_\ell|)}$$

and consequently,

$$\llbracket \underline{f}(M_1, \ldots M_\ell) \rrbracket^t * \langle \xi_1, \ldots, \xi_\ell \rangle = f \circ \langle f_1, \ldots, f_\ell \rangle \in \mathcal{PB}_{\iota}^{(n+|\Sigma_1|+\cdots+|\Sigma_\ell|)}$$

– The claim for conditionals follows from Lemma 2.25(ii).

– For applications suppose $\boldsymbol{\theta}, x_1 : \tau_1, \ldots, x_\ell : \tau_\ell \mid \Sigma_1 +\!\!+ \Sigma_2 +\!\!+ \Sigma_3 : \tau_\ell \vdash_{\mathrm{tr}} M\, N : \tau$ because

$$\begin{aligned} &\boldsymbol{\theta}, x_1 : \tau_1, \ldots, x_\ell : \tau_\ell \mid \Sigma_1 \vdash_{\mathrm{tr}} M : \tau' \bullet \Sigma_3 \to \tau \\ &\boldsymbol{\theta}, x_1 : \tau_1, \ldots, x_\ell : \tau_\ell \mid \Sigma_2 \vdash_{\mathrm{tr}} M\, N : \tau' \end{aligned}$$

Let $n \in \mathbb{N}$ and $\xi_1 \in \mathcal{PB}_{\tau_1}^{(n)}, \ldots, \xi_\ell \in \mathcal{PB}_{\tau_\ell}^{(n)}$. Define $\widehat{\xi}_i : \boldsymbol{\Theta} \times \mathbb{R}^{n+|\Sigma_1|} \to \llbracket \tau_i \rrbracket$ by $\widehat{\xi}_i(\boldsymbol{\theta}, \boldsymbol{z} +\!\!+ \boldsymbol{z}') \coloneqq \xi_i(\boldsymbol{\theta}, \boldsymbol{z})$. By the inductive hypothesis,

$$\llbracket M \rrbracket^t * \langle \xi_1, \ldots, \xi_\ell \rangle \in \mathcal{PB}_{\tau' \bullet \Sigma_3 \to \tau}^{(n+|\Sigma_1|)} \qquad \llbracket N \rrbracket^t * \langle \widehat{\xi}_1, \ldots, \widehat{\xi}_\ell \rangle \in \mathcal{PB}_{\tau'}^{(n+|\Sigma_1|+|\Sigma_2|)}$$

By definition of $\mathcal{PB}_{\tau' \bullet \Sigma_3 \to \tau}^{(n+|\Sigma_1|)}$,

$$(\llbracket M \rrbracket^t * \langle \xi_1, \ldots, \xi_\ell \rangle) \odot (\llbracket N \rrbracket^t * \langle \widehat{\xi}_1, \ldots, \widehat{\xi}_\ell \rangle) \in \mathcal{PB}_{\tau}^{(n+|\Sigma_1|+|\Sigma_2|+|\Sigma_3|)}$$

and by definition of $\odot$ and $*$,

$$(\llbracket M \rrbracket^t * \langle \xi_1, \ldots, \xi_\ell \rangle) \odot (\llbracket N \rrbracket^t * \langle \widehat{\xi}_1, \ldots, \widehat{\xi}_\ell \rangle) = \llbracket M\, N \rrbracket^t * \langle \xi_1, \ldots, \xi_\ell \rangle$$

– For abstractions suppose $\boldsymbol{\theta}, x_1 : \tau_1, \ldots, x_\ell : \tau_\ell \mid [] \vdash_{\mathrm{tr}} \lambda y.\, M : \tau \bullet \Sigma \to \tau'$ because $\boldsymbol{\theta}, x_1 : \tau_1, \ldots, x_\ell : \tau_\ell, y : \tau \mid \Sigma \vdash_{\mathrm{tr}} M : \tau'$; let $n \in \mathbb{N}$ and $\xi_1 \in \mathcal{PB}_{\tau_1}^{(n)}, \ldots, \xi_\ell \in \mathcal{PB}_{\tau_\ell}^{(n)}$.

To show the claim, suppose $n_2 \in \mathbb{N}$ and $g \in \mathcal{PB}_{\tau}^{(n+n_2)}$. By definition of the logical predicate we need to verify $\big(\llbracket \lambda y.\, M \rrbracket^t * \langle \xi_1, \ldots, \xi_\ell \rangle\big) \odot g \in \mathcal{PB}_{\tau'}^{(n+n_2+|\Sigma|)}$.

Call $\widehat{\xi}_i$ the extension of ξ_i to $\boldsymbol{\Theta} \times \mathbb{R}^{n+n_2} \to \mathbb{R}$. By the inductive hypothesis,

$$\llbracket M \rrbracket^t * \langle \widehat{\xi}_1, \ldots, \widehat{\xi}_\ell, g \rangle \in \mathcal{PB}_{\tau'}^{(n+n_2+|\Sigma|)}$$

Finally, it suffices to observe that

$$\big(\llbracket \lambda y.\, M \rrbracket^t * \langle \xi_1, \ldots, \xi_\ell \rangle\big) \odot g = \llbracket M \rrbracket^t * \langle \widehat{\xi}_1, \ldots, \widehat{\xi}_\ell, g \rangle$$

– Finally, $\llbracket \mathbf{sample}\, \mathcal{D} \rhd \varphi_{\boldsymbol{\theta}} \rrbracket^t * \langle \xi_1, \ldots, \xi_\ell \rangle(\boldsymbol{\theta}, \boldsymbol{z} +\!\!+ [z']) = z'$. □

4.5.1 More Permissible Type System

Assumption 4.17 (in conjunction with Assumption 4.16) is simple yet guarantees integrability. However, it is somewhat restrictive; in particular, it does not allow the expression of many ELBOs arising in variational inference as they often have the form of logarithms of exponential terms (cf. Example 4.6). For instance, $\sigma \mapsto \sigma^{-1}$ is not bounded by a polynomial and hence the normal log-pdf is also not bounded by a polynomial (unless the standard deviation is fixed). Therefore, we would like to also support the primitive operations log, exp and $^{-1}$ (inverses). These primitives are indispensable for expressing probability densities and the ELBO directly. For example Example 4.6 uses logarithms and densities involving exponentials and inverses.

Assumption 4.22. $\mathrm{Op} = \mathrm{Op}_{\mathrm{pb}} \cup \{\exp, \log, (-)^{-1}\}$, where $\mathrm{Op}_{\mathrm{pb}}$ is a set of smooth functions bounded by polynomials.

To maintain integrability we need to ensure that subterms involving $\underline{\exp}$ are "neutralised" by a corresponding $\underline{\log}$. To achieve this we annotate base types with $\mathbf{c}$, $\mathbf{p}$ or $\mathbf{l}$:

$$R^{(\mathbf{c})} \qquad R^{(\mathbf{p})} \qquad R^{(\mathbf{c})}_{>0} \qquad R^{(\mathbf{p})}_{>0} \qquad R^{(\mathbf{l})}_{>0}$$

Intuitively, $\mathbf{c}$ means "constant" (for fixed parameters), $\mathbf{p}$ means "integrable" and $\mathbf{l}$ means "needs to be passed through log". In particular there is *no* type $R^{(\mathbf{l})}$. More precisely, we constrain the typing rules, which are presented in Fig. 4.3c, such that

(i) if $\boldsymbol{\theta} \mid \Sigma \vdash_{\mathrm{int}} M : \iota^{(\mathbf{c})}$ then $[\![M]\!](\boldsymbol{\theta}, (-))$ is constant for fixed $\boldsymbol{\theta}$
(ii) if $\boldsymbol{\theta} \mid \Sigma \vdash_{\mathrm{int}} M : \iota^{(\mathbf{p})}$ then $[\![M]\!](\boldsymbol{\theta}, (-))$ is bounded by a polynomial for fixed $\boldsymbol{\theta}$
(iii) if $\boldsymbol{\theta} \mid \Sigma \vdash_{\mathrm{int}} M : R^{(\mathbf{l})}_{>0}$ then $\log([\![M]\!](\boldsymbol{\theta}, (-)))$ is bounded by a polynomial for fixed $\boldsymbol{\theta}$.

The subtyping relation is defined in Fig. 4.3b.

Motivated by the closure under composition (Lemma 2.25(i)), polynomially bounded primitives can be typed as $\iota^{(\mathbf{p})}$ provided the arguments are also annotated with $\mathbf{p}$. Furthermore, *all* primitives preserve constant terms (for fixed $\boldsymbol{\theta} \in \boldsymbol{\Theta}$):

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\mathrm{int}} M_1 : \iota_1^{(\mathbf{c})} \quad \cdots \quad \Gamma \mid \Sigma_\ell \vdash_{\mathrm{int}} M_\ell : \iota_\ell^{(\mathbf{c})}}{\Gamma \mid \Sigma_1 +\!\!+ \cdots +\!\!+ \Sigma_\ell \vdash_{\mathrm{int}} \underline{f}(M_1, \dots M_\ell) : \iota^{(\mathbf{c})}} \; f : [\![\iota_1]\!] \times \cdots \times [\![\iota_\ell]\!] \to [\![\iota]\!] \in \mathrm{Op}$$

$\underline{\exp}$ and $\underline{\log}$ alter the annotation in the expected manner:

$$\frac{\Gamma \mid \Sigma \vdash_{\mathrm{int}} M : R^{(\mathbf{p})}}{\Gamma \mid \Sigma \vdash_{\mathrm{int}} \underline{\exp}(M) : R^{(\mathbf{l})}_{>0}} \qquad \frac{\Gamma \mid \Sigma \vdash_{\mathrm{int}} M : R^{(\mathbf{l})}_{>0}}{\Gamma \mid \Sigma \vdash_{\mathrm{int}} \underline{\log}(M) : R^{(\mathbf{p})}}$$

The rules

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\mathrm{int}} M_1 : R^{(\mathbf{l})}_{>0} \quad \Gamma \mid \Sigma_2 \vdash_{\mathrm{int}} M_2 : R^{(\mathbf{l})}_{>0}}{\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 \vdash_{\mathrm{int}} M_1 \underline{\cdot} M_2 : R^{(\mathbf{l})}_{>0}} \qquad \frac{\Gamma \mid \Sigma \vdash_{\mathrm{int}} M : R^{(\mathbf{l})}_{>0}}{\Gamma \mid \Sigma \vdash_{\mathrm{int}} \underline{M^{-1}} : R^{(\mathbf{l})}_{>0}}$$

are motivated by the elementary property that $\log \circ (f \cdot g) = (\log \circ f) + (\log \circ g)$ and $\log \circ (f^{-1}) = -\log \circ f$ for $f, g : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$.

trace types $\quad \Sigma ::= [\mathcal{D}_1, \ldots, \mathcal{D}_n] \qquad n \geq 0$

base types $\quad \iota ::= R \mid R_{>0}$

types $\quad \tau ::= R^{(\mathbf{c})} \mid R^{(\mathbf{p})} \mid R^{(\mathbf{c})}_{>0} \mid R^{(\mathbf{p})}_{>0} \mid R^{(\mathbf{l})}_{>0} \mid \tau \bullet \Sigma \to \tau$

(a) Types

$$\frac{}{\iota_1^{(\mathbf{c})} \sqsubseteq_{\text{int}} \iota_2^{(\mathbf{c})}}\ \iota_1 \sqsubseteq \iota_2 \qquad \frac{}{\iota_1^{(\mathbf{p})} \sqsubseteq_{\text{int}} \iota_2^{(\mathbf{p})}}\ \iota_1 \sqsubseteq \iota_2 \qquad \frac{}{R^{(\mathbf{l})}_{>0} \sqsubseteq_{\text{int}} R^{(\mathbf{l})}_{>0}}$$

$$\frac{}{\iota_1^{(\mathbf{c})} \sqsubseteq_{\text{int}} \iota_2^{(\mathbf{p})}}\ \iota_1 \sqsubseteq \iota_2 \qquad \frac{}{R^{(\mathbf{c})}_{>0} \sqsubseteq_{\text{int}} R^{(\mathbf{l})}_{>0}}$$

$$\frac{\tau_1' \sqsubseteq_{\text{int}} \tau_1 \quad \tau_2 \sqsubseteq_{\text{int}} \tau_2'}{(\tau_1 \bullet \Sigma \to \tau_2) \sqsubseteq_{\text{int}} (\tau_1' \bullet \Sigma \to \tau_2')}$$

(b) Subtyping

$$\frac{\Gamma \mid \Sigma \vdash_{\text{int}} M : \tau}{\Gamma \mid \Sigma \vdash_{\text{int}} M : \tau'}\ \tau \sqsubseteq_{\text{int}} \tau' \qquad \frac{}{x_1 : \tau_1, \ldots, x_m : \tau_m \mid [] \vdash_{\text{int}} x_i : \tau_i}$$

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\text{int}} M_1 : \iota_1^{(\mathbf{p})} \quad \cdots \quad \Gamma \mid \Sigma_\ell \vdash_{\text{int}} M_\ell : \iota_\ell^{(\mathbf{p})}}{\Gamma \mid \Sigma_1 +\!\!+ \cdots +\!\!+ \Sigma_\ell \vdash_{\text{int}} \underline{f}(M_1, \ldots M_\ell) : \iota^{(\mathbf{p})}}\ f : [\![\iota_1]\!] \times \cdots \times [\![\iota_\ell]\!] \to [\![\iota]\!] \in \text{Op}_{\text{pb}}$$

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\text{int}} M_1 : \iota_1^{(\mathbf{c})} \quad \cdots \quad \Gamma \mid \Sigma_\ell \vdash_{\text{int}} M_\ell : \iota_\ell^{(\mathbf{c})}}{\Gamma \mid \Sigma_1 +\!\!+ \cdots +\!\!+ \Sigma_\ell \vdash_{\text{int}} \underline{f}(M_1, \ldots M_\ell) : \iota^{(\mathbf{c})}}\ f : [\![\iota_1]\!] \times \cdots \times [\![\iota_\ell]\!] \to [\![\iota]\!] \in \text{Op}$$

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\text{int}} M_1 : R^{(\mathbf{l})}_{>0} \quad \Gamma \mid \Sigma_2 \vdash_{\text{int}} M_2 : R^{(\mathbf{l})}_{>0}}{\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 \vdash_{\text{int}} M_1 \mathbin{\underline{\cdot}} M_2 : R^{(\mathbf{l})}_{>0}} \qquad \frac{\Gamma \mid \Sigma \vdash_{\text{int}} M : R^{(\mathbf{l})}_{>0}}{\Gamma \mid \Sigma \vdash_{\text{int}} M^{\underline{-1}} : R^{(\mathbf{l})}_{>0}}$$

$$\frac{\Gamma \mid \Sigma \vdash_{\text{int}} M : R^{(\mathbf{p})}}{\Gamma \mid \Sigma \vdash_{\text{int}} \underline{\exp}(M) : R^{(\mathbf{l})}_{>0}} \qquad \frac{\Gamma \mid \Sigma \vdash_{\text{int}} M : R^{(\mathbf{l})}_{>0}}{\Gamma \mid \Sigma \vdash_{\text{int}} \underline{\log}(M) : R^{(\mathbf{p})}}$$

$$\frac{\Gamma \mid \Sigma \vdash_{\text{int}} L : R^{(\mathbf{p})} \quad \Gamma \mid \Sigma' \vdash_{\text{int}} M : \iota^{(\mathbf{p})} \quad \Gamma \mid \Sigma'' \vdash_{\text{int}} N : \iota^{(\mathbf{p})}}{\Gamma \mid \Sigma +\!\!+ \Sigma' +\!\!+ \Sigma'' \vdash_{\text{int}} \textbf{if}\, L < 0\, \textbf{then}\, M\, \textbf{else}\, N : \iota^{(\mathbf{p})}}$$

$$\frac{\Gamma, y : \tau_1 \mid \Sigma \vdash_{\text{int}} M : \tau_2}{\Gamma \mid [] \vdash_{\text{int}} \lambda y.\, M : \tau_1 \bullet \Sigma \to \tau_2}$$

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\text{int}} M : \tau_1 \bullet \Sigma_3 \to \tau_2 \quad \Gamma \mid \Sigma_2 \vdash_{\text{int}} N : \tau_1}{\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 +\!\!+ \Sigma_3 \vdash_{\text{int}} M\, N : \tau_2}$$

$$\frac{}{\boldsymbol{\theta} \mid [\mathcal{D}] \vdash_{\text{int}} \textbf{sample}_{\mathcal{D}} \rhd \varphi_{\boldsymbol{\theta}} : R^{(\mathbf{p})}}\ \varphi_{(-)} \in \text{Diff}_{\boldsymbol{\Theta}}(\text{supp}(\mathcal{D}))$$

(c) Typing rules for $\vdash_{\text{int}}$

Figure 4.3: A more permissible type system

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \mid [] \vdash_{\text{int}} \sqrt{2\pi} : R^{(\mathbf{c})}_{>0} \qquad \Gamma \mid [] \vdash_{\text{int}} s : R^{(\mathbf{c})}_{>0}}{\Gamma \mid [] \vdash_{\text{int}} \sqrt{2\pi} \cdot s : R^{(\mathbf{c})}_{>0}}}{\Gamma \mid [] \vdash_{\text{int}} \left(\sqrt{2\pi} \cdot s\right)^{-1} : R^{(\mathbf{c})}_{>0}}}{\Gamma \mid [] \vdash_{\text{int}} \left(\sqrt{2\pi} \cdot s\right)^{-1} : R^{(\mathbf{l})}_{>0}} \qquad \dfrac{\star}{\Gamma \mid [] \vdash_{\text{int}} \exp\left(-0.5 \cdot \left((x-m) \cdot s^{-1}\right)^2\right) : R^{(\mathbf{l})}_{>0}}}{\Gamma \mid [] \vdash_{\text{int}} \left(\sqrt{2\pi} \cdot s\right)^{-1} \cdot \exp\left(-0.5 \cdot \left((x-m) \cdot s^{-1}\right)^2\right) : R^{(\mathbf{l})}_{>0}}}{x : R^{(\mathbf{p})}, m : R^{(\mathbf{p})} \mid [] \vdash_{\text{int}} \lambda s. \left(\sqrt{2\pi} \cdot s\right)^{-1} \cdot \exp\left(-0.5 \cdot \left((x-m) \cdot s^{-1}\right)^2\right) : R^{(\mathbf{c})}_{>0} \to R^{(\mathbf{l})}_{>0}}}{x : R^{(\mathbf{p})} \mid [] \vdash_{\text{int}} \lambda m, s. \left(\sqrt{2\pi} \cdot s\right)^{-1} \cdot \exp\left(-0.5 \cdot \left((x-m) \cdot s^{-1}\right)^2\right) : R^{(\mathbf{p})} \to R^{(\mathbf{c})}_{>0} \to R^{(\mathbf{l})}_{>0}}}{\mid [] \vdash_{\text{int}} \lambda x, m, s. \left(\sqrt{2\pi} \cdot s\right)^{-1} \cdot \exp\left(-0.5 \cdot \left((x-m) \cdot s^{-1}\right)^2\right) : R^{(\mathbf{p})} \to R^{(\mathbf{p})} \to R^{(\mathbf{c})}_{>0} \to R^{(\mathbf{l})}_{>0}}$$

(a) Derivation of the typing judgment of Example 4.23, where $\Gamma \coloneqq x : R^{(\mathbf{p})}, m : R^{(\mathbf{p})}, s : R^{(\mathbf{c})}_{>0}$. The derivation $\star$ is continued in Fig. 4.4b.

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \mid [] \vdash_{\text{int}} x : R^{(\mathbf{p})} \qquad \Gamma \mid [] \vdash_{\text{int}} m : R^{(\mathbf{p})}}{\Gamma \mid [] \vdash_{\text{int}} x - m : R^{(\mathbf{p})}} \qquad \dfrac{\dfrac{\Gamma \mid [] \vdash_{\text{int}} s : R^{(\mathbf{c})}_{>0}}{\Gamma \mid [] \vdash_{\text{int}} s^{-1} : R^{(\mathbf{c})}_{>0}}}{\Gamma \mid [] \vdash_{\text{int}} s^{-1} : R^{(\mathbf{p})}}}{\Gamma \mid [] \vdash_{\text{int}} (x-m) \cdot s^{-1} : R^{(\mathbf{p})}}}{\Gamma \mid [] \vdash_{\text{int}} \left((x-m) \cdot s^{-1}\right)^2 : R^{(\mathbf{p})}} \qquad \Gamma \mid [] \vdash_{\text{int}} -0.5 : R^{(\mathbf{p})}}{\Gamma \mid [] \vdash_{\text{int}} -0.5 \cdot \left((x-m) \cdot s^{-1}\right)^2 : R^{(\mathbf{p})}}}{\Gamma \mid [] \vdash_{\text{int}} \exp\left(-0.5 \cdot \left((x-m) \cdot s^{-1}\right)^2\right) : R^{(\mathbf{l})}_{>0}}$$

(b) Continuation of the derivation (derivation of $\star$).

Figure 4.4: Derivation of the typing judgment of Example 4.23.

Note that the branches of conditionals cannot have type $R^{(\mathbf{l})}_{>0}$. This is motivated by the smoothed interpretation, which we will develop in the next chapter. This uses addition in the interpretation of conditionals which does not behave nicely when composed with logarithms.

As the following example illustrates, the type system allows us to encode normal probability density functions, which are required for applications to variational inference:

Example 4.23. For the Variational Inference Example 4.6,

$$\mid [] \vdash_{\text{int}} \lambda x, m, s. \left(\sqrt{2\pi} \cdot s\right)^{-1} \cdot \exp\left(-0.5 \cdot \left((x-m) \cdot s^{-1}\right)^2\right) : R^{(\mathbf{p})} \to R^{(\mathbf{p})} \to R^{(\mathbf{c})}_{>0} \to R^{(\mathbf{l})}_{>0}$$

This typing judgement is derived in Fig. 4.4.

Type Soundness

To formally establish type soundness, we can use a logical predicate $\mathcal{PB}^{(n)}_{\tau}$ on $\mathbf{\Theta} \times \mathbb{R}^n \to [\![\tau]\!]$, which is very similar to the one defined earlier in this section:

Table 4.1: Overview of type systems in this thesis.

Property	Section	Judgement	Annotation
–	Section 2.2.1	$\vdash$	–
totality	Section 4.1	$\vdash_{\mathrm{tr}}$	none/$*$
integrability	Section 4.5	$\vdash_{\mathrm{tr}}$ (+ Ass. 4.17)	none/$*$
	Section 4.5.1	$\vdash_{\mathrm{int}}$	$\mathbf{c}/\mathbf{p}/\mathbf{l}$
uniform convergence	Section 5.3.2	$\vdash_{\mathrm{unif}}$	$(\mathbf{f},\mathrm{dep})/(\mathbf{t},\mathrm{dep})$
affine guards	Section 8.4	$\vdash_{\mathrm{aff}}$	$\mathbf{f}/\mathbf{t}$

(i) $f \in \mathcal{PB}^{(n)}_{\iota(\mathbf{c})}$ if for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, $f(\boldsymbol{\theta}, (-))$ is constant

(ii) $f \in \mathcal{PB}^{(n)}_{\iota(\mathbf{p})}$ if for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, $f(\boldsymbol{\theta}, (-))$ is bounded by a polynomial

(iii) $f \in \mathcal{PB}^{(n)}_{R^{(\mathbf{l})}_{>0}}$ if for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, $\log(f(\boldsymbol{\theta}, (-)))$ is bounded by a polynomial

(iv) $f \in \mathcal{PB}^{(n)}_{\tau_1 \bullet \Sigma_3 \to \tau_2}$ if for all $n_2 \in \mathbb{N}$ and $g \in \mathcal{PB}^{(n+n_2)}_{\tau_1}$, $f \odot g \in \mathcal{PB}^{(n+n_2+|\Sigma_3|)}_{\tau_2}$

where for $f : \mathbb{R}^{n_1} \to [\![\tau_1 \bullet \Sigma_3 \to \tau_2]\!]$ and $g : \mathbb{R}^{n_1+n_2} \to [\![\tau_1]\!]$ we define

$$\begin{aligned} f \odot g : \mathbb{R}^{n_1+n_2+|\Sigma_3|} &\to \tau_2 \\ \boldsymbol{s}_1 +\!\!+ \boldsymbol{s}_2 +\!\!+ \boldsymbol{s}_3 &\mapsto f(\boldsymbol{s}_1)(g(\boldsymbol{s}_1 +\!\!+ \boldsymbol{s}_2), \boldsymbol{s}_3) \end{aligned}$$

The following is immediate:

Lemma 4.24. *If* $f \in \mathcal{PB}^{(n)}_{\tau}$ *and* $\tau \sqsubseteq_{\mathrm{int}} \tau'$ *then*[6] $\mathrm{emb}_{\tau,\tau'}(f) \in \mathcal{PB}^{(n)}_{\tau'}$.

The following Fundamental Lemma is proven by an induction on the typing derivation as for Lemma 4.20. The rationales for the rules described above provide the justifications for the additional rules for the primitives.

Lemma 4.25 (Fundamental)**.** *If* $\theta_1 : \iota_1^{(\mathbf{c})}, \ldots, \theta_m : \iota_m^{(\mathbf{c})}, x_1 : \tau_1, \ldots, x_\ell : \tau_\ell \mid \Sigma \vdash_{\mathrm{int}} M : \tau$, $n \in \mathbb{N}$ *and* $\xi_1 \in \mathcal{PB}^{(n)}_{\tau_1}, \ldots, \xi_\ell \in \mathcal{PB}^{(n)}_{\tau_\ell}$ *then*

$$[\![M]\!] * \langle \xi_1, \ldots, \xi_\ell \rangle \in \mathcal{PB}^{(n+|\Sigma|)}_{\tau}$$

where $([\![M]\!] * \langle \xi_1, \ldots, \xi_\ell \rangle(\boldsymbol{\theta}, \boldsymbol{s} +\!\!+ \boldsymbol{s}')) \coloneqq [\![M]\!](\boldsymbol{\theta}, (\xi_1(\boldsymbol{s}), \ldots, \xi_\ell(\boldsymbol{s})), \boldsymbol{s}')$.

We conclude as for Proposition 4.21:

Proposition 4.26 (Integrability)**.** *Let* $\theta_1 : \iota_1^{(\mathbf{c})}, \ldots, \theta_m : \iota_m^{(\mathbf{c})} \mid \Sigma \vdash_{\mathrm{int}} M : R^{(\mathbf{p})}$. *For all* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, $\mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}}[[\![M]\!](\boldsymbol{\theta}, \boldsymbol{s})] < \infty$.

Consequently, the Optimisation Problem 4.12 is well-defined.

4.6 Generic Type System with Annotations

In the preceding section we have designed type systems to obtain stronger guarantees for terms $\boldsymbol{\theta} \mid \Sigma \vdash_{\mathrm{tr}} M : R$ beyond measurability by annotating base types. We conclude

[6] where emb is defined as in Section 4.2 ignoring the annotation

$$\frac{\Gamma \mid \Sigma \vdash_? M : \tau}{\Gamma \mid \Sigma \vdash_? M : \tau'}\ \tau \sqsubseteq_? \tau' \qquad \frac{}{x_1 : \tau_1, \ldots, x_m : \tau_m \mid [] \vdash_? x_i : \tau_i}$$

$$\frac{\Gamma \mid \Sigma_1 \vdash_? M_1 : \iota_1^{(\spadesuit_1)} \quad \cdots \quad \Gamma \mid \Sigma_\ell \vdash_? M_\ell : \iota_\ell^{(\spadesuit_\ell)}}{\Gamma \mid \Sigma_1 +\!\!+ \cdots +\!\!+ \Sigma_\ell \vdash_? \underline{f}(M_1, \ldots M_\ell) : \iota^{(\spadesuit)}}\ f : [\![\iota_1]\!] \times \cdots \times [\![\iota_\ell]\!] \to [\![\iota]\!] \in \mathrm{Op},\ f, \spadesuit_1, \ldots, \spadesuit_\ell, \spadesuit \text{ satisfy } \ldots$$

$$\frac{\Gamma \mid \Sigma \vdash_? L : R^{(\spadesuit')} \quad \Gamma \mid \Sigma' \vdash_? M : \iota^{(\spadesuit)} \quad \Gamma \mid \Sigma'' \vdash_? N : \iota^{(\spadesuit)}}{\Gamma \mid \Sigma +\!\!+ \Sigma' +\!\!+ \Sigma'' \vdash_? \mathbf{if}\ L < 0\ \mathbf{then}\ M\ \mathbf{else}\ N : \iota^{(\spadesuit)}}\ \spadesuit \text{ and } \spadesuit' \text{ satisfy } \ldots$$

$$\frac{\Gamma, y : \tau_1 \mid \Sigma \vdash_? M : \tau_2}{\Gamma \mid [] \vdash_? \lambda y.\, M : \tau_1 \bullet \Sigma \to \tau_2}$$

$$\frac{\Gamma \mid \Sigma_1 \vdash_? M : \tau_1 \bullet \Sigma_3 \to \tau_2 \quad \Gamma \mid \Sigma_2 \vdash_? N : \tau_1}{\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 +\!\!+ \Sigma_3 \vdash_? M\, N : \tau_2}$$

$$\frac{}{\boldsymbol{\theta} \mid [\mathcal{D}] \vdash_? \mathbf{sample}_{\mathcal{D}} \rhd \varphi_{\boldsymbol{\theta}} : R^{(\spadesuit)}}\ \varphi_{(-)} \in \mathrm{Diff}_{\boldsymbol{\Theta}}(\mathrm{supp}(\mathcal{D})),\ \spadesuit \text{ satisfies } \ldots$$

Figure 4.5: Generic type system with annotations.

the chapter by presenting a generic type system following this approach with judgements of the form $\Gamma \mid \Sigma \vdash_? M : \tau$, where "?" indicates the property we aim to establish. Throughout this thesis we give manifold instantiations of this scheme; see Table 4.1 for an overview.

Types are generated from

$$\begin{array}{ll} \textbf{trace types} & \Sigma ::= [\mathcal{D}_1, \ldots, \mathcal{D}_n] \\ \textbf{base types} & \iota ::= R \mid R_{>0} \\ \textbf{types} & \tau ::= R^{(\spadesuit)} \mid R_{>0}^{(\spadesuit')} \mid \tau \bullet \Sigma \to \tau \end{array}$$

where annotations $\spadesuit$ and $\spadesuit'$ are drawn from two sets (which do not need to be disjoint). Thus, we can ensure that some annotation cannot be both used for R and $R_{>0}$. (Recall that in Section 4.5.1 we ruled out $R^{(\mathbf{1})}$.) Furthermore, we assume a partial order $\sqsubseteq_?$ on base types such that $\iota_1^{(\spadesuit)} \sqsubseteq_? \iota_2^{(\spadesuit')}$ only if $\iota_1 \sqsubseteq \iota_2$, which is lifted to higher orders as before:

$$\frac{\tau_1' \sqsubseteq_? \tau_1 \quad \tau_2 \sqsubseteq_? \tau_2'}{(\tau_1 \bullet \Sigma \to \tau_2) \sqsubseteq_? (\tau_1' \bullet \Sigma \to \tau_2')} \tag{4.2}$$

The typing rules have the same form but they are extended with the annotations on base types and side conditions possibly constraining them. The rules for subtyping, variables, abstractions and applications do not need to be changed at all, but they use annotated types instead of the types of Section 4.1. The full type system is presented in Fig. 4.5.

$\vdash_{\mathrm{tr}}$ can be considered a special case of $\vdash_?$ whereby we use the singleton $*$ as annotations and no (additional) side conditions; for $\vdash_{\mathrm{int}}$ we use the annotations $\mathbf{c}/\mathbf{p}/\mathbf{l}$ and impose some additional restrictions e.g. that exp requires an argument of type $R^{(\mathbf{p})}$.

Finally, $\vdash_?$ and its instantiations refine the basic type system $\vdash_{\mathrm{tr}}$ of Section 4.1 in the sense that if a term-in-context is provable in the annotated type system, then its erasure (i.e. erasure of the annotations of base types and distributions) is provable in the basic type system. This is straightforward to check.

4.7 Conclusion and Related Work

We have presented a (probabilistic) programming language framework to pose stochastic optimisation problems arising e.g. in variational inference. We have employed trace types to precisely capture samples and their distributions, and we have endowed our language with a denotational (measurable) value semantics. Finally, we have discussed assumptions and a type system to guarantee that the objective function is well-defined.

Related Work

Motivated by guaranteeing absolute continuity (which is a necessary but not sufficient criterion for the correctness of e.g. variational inference), [Lew et al., 2020] use an approach similar to our trace types to track the samples which are drawn. Whilst we gather the distributions of samples (since we are interested in expectations), they collect only their support. Furthermore, they do not support standard conditionals but their "workaround" is also eager in the sense of combining the traces of both branches. Moreover, they do not support a full higher-order language in which higher-order terms can draw samples. Thus, they do not need to consider *function* types tracking the samples drawn during evaluation.

[Lee et al., 2020b] are one of the first to study the correctness of variational inference for (imperative) probabilistic programming. Using abstract interpretation [Cousot and Cousot, 1977] they verify some of the conditions required for the unbiasedness of the *Score* estimator. (In particular, they state a variant of Lemma 2.17 but they are not concerned with the *overall* correctness of the variational inference pipeline, which includes convergence of stochastic gradient descent.) They do ensure integrability of the objective function by focussing on normal distributions only and by making an assumption about the shape of the integrant. However, they do not fully discuss how to verify this.

Logical relations are a classic proof technique, which have also been used recently in probabilistic and differentiable programming (e.g. [Barthe et al., 2020a, Huot et al., 2020, Brunel et al., 2020, Mazza and Pagani, 2021, Lew et al., 2023]).

Chapter 5

Smoothing and Stochastic Gradient Descent

In the previous chapter we have posed an abstract optimisation problem generalising variational inference. Unfortunately, the approach which is most effective in practice, stochastic gradient descent with the reparameterisation gradient estimator, may be incorrect in the presence of conditionals: we can easily express the counter-Example 2.16 as

$$\theta : R \mid [\mathcal{N}] \vdash_{\mathrm{tr}} -0.5 \cdot \theta^2 + \mathbf{if}\,(\mathbf{sample}_{\mathcal{N}} \rhd \varphi_\theta) < 0\,\mathbf{then}\,0\,\mathbf{else}\,1 : R$$

where $\varphi_\theta(s) \coloneqq s + \theta$.

In this chapter, we provide a method for obtaining unbiased reparameterisation estimators. Our approach is based on a denotational semantics $[\![(-)]\!]_\eta$ (for accuracy coefficient $\eta > 0$) of programs in the cartesian closed category **Fr** of Frölicher spaces (see e.g. [Frölicher and Kriegl, 1988, Stacey, 2011]), which generalises smooth manifolds.

Intuitively, we replace the Heaviside step-function usually arising in the interpretation of conditionals by smooth approximations. In particular, we interpret the above conditional as

$$[\![\mathbf{if}\,(\mathbf{sample}_{\mathcal{N}} \rhd \varphi_\theta) < 0\,\mathbf{then}\,0\,\mathbf{else}\,1]\!]_\eta(\theta, s) \coloneqq \sigma_\eta(\varphi_\theta(s))$$

where σ_η is a smooth function. Thus, the program M is interpreted by a smooth function $[\![M]\!]_\eta$, for which the reparameterisation gradient may be estimated unbiasedly.

After having discussed the smoothed interpretation and having posed a smoothed variant of Problem 4.12 in Section 5.1, we show correctness of stochastic gradient descent with the reparameterisation gradient estimator for the smoothing in Section 5.2. Finally, we investigate the relation between the original and the smoothed optimisation problem (Section 5.3), and we propose a type system guaranteeing uniform convergence.

5.1 Smoothed Denotational Value Semantic

Now we turn to our smoothed denotational value semantics, which we use to circumvent the bias in the reparameterisation gradient estimator. It is parameterised by a family of smooth functions $\sigma_\eta : \mathbb{R} \to [0, 1]$ for $\eta > 0$.

(a) σ_η (black, dashed/dotted) and the Heaviside step function (red, solid)

(b) Derivatives of σ_η

Figure 5.1: Sigmoid function σ_η and its derivative (dotted: $\eta = 1/3$, dashed: $\eta = 1/15$).

Assumption 5.1. For $\eta > 0$, $\sigma_\eta : \mathbb{R} \to [0, 1]$ is smooth.

Assumption 5.2. Primitive operations in Op are smooth.

Example 5.3. Our primary example is $\sigma_\eta(x) \coloneqq \sigma(\frac{x}{\eta})$, where σ is the (logistic) sigmoid function $\sigma(x) \coloneqq \frac{1}{1+\exp(-x)}$, see Fig. 5.1a.

Whilst at this stage no further properties other than smoothness are required, we will later need to restrict σ_η to have good properties, in particular convergence to the Heaviside step function to recover the standard semantics of Section 4.2.

As a categorical model we use *Frölicher spaces*, which generalise smooth spaces.

5.1.1 Frölicher Spaces

Let $C^\infty(\mathbb{R}, \mathbb{R})$ be the set of smooth functions $\mathbb{R} \to \mathbb{R}$.

Definition 5.4. A ***Frölicher space*** is a triple $(X, \mathcal{C}_X, \mathcal{F}_X)$, where X is a set, $\mathcal{C}_X \subseteq \mathbf{Set}(\mathbb{R}, X)$ is a set of *curves* and $\mathcal{F}_X \subseteq \mathbf{Set}(X, \mathbb{R})$ is a set of *functionals* satisfying

(i) if $c \in \mathcal{C}_X$ and $f \in \mathcal{F}_X$ then $f \circ c \in C^\infty(\mathbb{R}, \mathbb{R})$
(ii) if $c : \mathbb{R} \to X$ such that for all $f \in \mathcal{F}_X$, $f \circ c \in C^\infty(\mathbb{R}, \mathbb{R})$ then $c \in \mathcal{C}_X$
(iii) if $f : X \to \mathbb{R}$ such that for all $c \in \mathcal{C}_X$, $f \circ c \in C^\infty(\mathbb{R}, \mathbb{R})$ then $f \in \mathcal{F}_X$.

A *morphism* between Frölicher spaces $(X, \mathcal{C}_X, \mathcal{F}_X)$ and $(Y, \mathcal{C}_Y, \mathcal{F}_Y)$ is a map $\varphi : X \to Y$ satisfying $f \circ \varphi \circ c \in C^\infty(\mathbb{R}, \mathbb{R})$ for all $f \in \mathcal{F}_Y$ and $c \in \mathcal{C}_X$.

Frölicher spaces and their morphisms constitute a category **Fr**, which is well-known to be cartesian closed [Frölicher and Kriegl, 1988, Stacey, 2011].

The categorical product of two Frölicher spaces $(X_1, \mathcal{C}_{X_1}, \mathcal{F}_{X_1})$ and $(X_2, \mathcal{C}_{X_2}, \mathcal{F}_{X_2})$ is $(X_1 \times X_2, \mathcal{C}_{X_1 \times X_2}, \mathcal{F}_{X_1 \times X_2})$, where

$$\mathcal{F}_{X_1 \times X_2} \coloneqq \{f : X_1 \times X_2 \to \mathbb{R} \mid \forall c_1 \in \mathcal{C}_{X_1}, c_2 \in \mathcal{C}_{X_2}.\, f \circ \langle c_1, c_2 \rangle \in C^\infty(\mathbb{R}, \mathbb{R})\}$$
$$\mathcal{C}_{X_1 \times X_2} \coloneqq \{c : \mathbb{R} \to X_1 \times X_2 \mid \forall f \in \mathcal{F}_{X_1 \times X_2}.\, f \circ c \in C^\infty(\mathbb{R}, \mathbb{R})\}$$

In particular, to verify that $\pi_i : X_1 \times X_2 \to X_i$ is a morphism it suffices to verify that for all $f \in \mathcal{F}_{X_i}$, $f \circ \pi_i \in \mathcal{F}_{X_1 \times X_2}$. To establish this, let $c_1 \in \mathcal{C}_{X_1}$ and $c_2 \in \mathcal{C}_{X_2}$. Then $f \circ \pi_i \circ \langle c_1, c_2 \rangle = f \circ c_i \in C^\infty(\mathbb{R}, \mathbb{R})$.

$$[\![\Gamma \mid \Sigma \vdash_{\mathrm{tr}} M : \tau']\!]_\eta(\gamma, \boldsymbol{s}) \coloneqq \mathrm{emb}_{\tau,\tau'}([\![\Gamma \mid \Sigma \vdash_{\mathrm{tr}} M : \tau]\!]_\eta(\gamma, \boldsymbol{s}))$$

$$[\![x_1 : \tau_1, \dots, x_m : \tau_m \mid [] \vdash_{\mathrm{tr}} x_i : \tau_i]\!]_\eta(\gamma, []) \coloneqq \gamma_i$$

$$[\![\Gamma \mid \Sigma_1 +\!\!+ \cdots +\!\!+ \Sigma_\ell \vdash_{\mathrm{tr}} \underline{f}(M_1, \dots, M_\ell) : \iota]\!]_\eta(\gamma, \boldsymbol{s}_1 +\!\!+ \cdots +\!\!+ \boldsymbol{s}_\ell) \coloneqq f([\![\Gamma \mid \Sigma_1 \vdash_{\mathrm{tr}} M_1 : \iota_1]\!]_\eta(\gamma, \boldsymbol{s}_1), \dots, [\![\Gamma \mid \Sigma_\ell \vdash_{\mathrm{tr}} M_\ell : \iota_\ell]\!]_\eta(\gamma, \boldsymbol{s}_\ell))$$

$$[\![\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 +\!\!+ \Sigma_3 \vdash_{\mathrm{tr}} M\, N : \tau]\!]_\eta(\gamma, \boldsymbol{s}_1 +\!\!+ \boldsymbol{s}_2 +\!\!+ \boldsymbol{s}_3) \coloneqq [\![\Gamma \mid \Sigma_1 \vdash_{\mathrm{tr}} M : \tau_1 \bullet \Sigma_3 \to \tau_2]\!]_\eta(\gamma, \boldsymbol{s}_1)([\![\Gamma \mid \Sigma_2 \vdash_{\mathrm{tr}} N : \tau_1]\!]_\eta(\gamma, \boldsymbol{s}_2), \boldsymbol{s}_3)$$

$$[\![\Gamma \mid [] \vdash_{\mathrm{tr}} \lambda y.\, M : \tau_1 \bullet \Sigma \to \tau_2]\!]_\eta(\gamma, []) \coloneqq (v, \boldsymbol{s}) \in [\![\tau_1]\!]_\eta \times [\![\Sigma]\!]_\eta \mapsto [\![\Gamma, x : \tau_1 \mid \Sigma \vdash_{\mathrm{tr}} M : \tau_2]\!]_\eta((\gamma, v), \boldsymbol{s})$$

$$\begin{aligned}[\![\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 +\!\!+ \Sigma_3 \vdash_{\mathrm{tr}} \mathbf{if}\, L < 0\, \mathbf{then}\, M\, \mathbf{else}\, N : \iota]\!]_\eta(\gamma, \boldsymbol{s}_1 +\!\!+ \boldsymbol{s}_2 +\!\!+ \boldsymbol{s}_3)) \coloneqq \\ \sigma_\eta(-[\![\Gamma \mid \Sigma_1 \vdash_{\mathrm{tr}} L : R]\!]_\eta(\gamma, \boldsymbol{s}_1)) \cdot [\![\Gamma \mid \Sigma_2 \vdash_{\mathrm{tr}} M : \tau]\!]_\eta(\gamma, \boldsymbol{s}_2) \\ + \sigma_\eta([\![\Gamma \mid \Sigma_1 \vdash_{\mathrm{tr}} L : R]\!]_\eta(\gamma, \boldsymbol{s}_1)) \cdot [\![\Gamma \mid \Sigma_3 \vdash_{\mathrm{tr}} N : \tau]\!]_\eta(\gamma, \boldsymbol{s}_3)\end{aligned}$$

$$[\![\boldsymbol{\theta}, \Gamma \mid [\mathcal{D}] \vdash_{\mathrm{tr}} \mathbf{sample}\, \mathcal{D} \rhd \varphi_{\boldsymbol{\theta}} : R]\!]_\eta((\boldsymbol{\theta}, \gamma), [s]) \coloneqq \varphi_{\boldsymbol{\theta}}(s)$$

Figure 5.2: η-Smoothed interpretation of terms-in-context $[\![\Gamma \mid \Sigma \vdash_{\mathrm{tr}} M : \tau]\!]_\eta : [\![\Gamma]\!]_\eta \times [\![\Sigma]\!]_\eta \to [\![\tau]\!]_\eta$, where $\eta > 0$.

$\mathbb{R}$ can be endowed with a Frölicher space structure by stipulating $\mathcal{C}_\mathbb{R} \coloneqq \mathcal{F}_\mathbb{R} \coloneqq C^\infty(\mathbb{R}, \mathbb{R})$. Likewise, $(\mathbb{R}_{>0}, C^\infty(\mathbb{R}, \mathbb{R}_{>0}), C^\infty(\mathbb{R}_{>0}, \mathbb{R}))$ is a Frölicher space.

Remark 5.5. (i) If $\varphi : X_1 \times \cdots \times X_n \to X$ is smooth, where $X_1, \dots, X_n, X \in \{\mathbb{R}, \mathbb{R}_{>0}\}$, then φ is also a morphism. For this it suffices to establish that for all $f \in \mathcal{F}_X$, $f \circ \varphi \in \mathcal{F}_{X_1 \times \cdots \times X_n}$. To verify this, consider, $c_1 \in \mathcal{C}_{X_1}, \dots, c_n \in \mathcal{C}_{X_n}$ and note that $f \circ \varphi \circ \langle c_1, \dots, c_n \rangle \in C^\infty(\mathbb{R}, \mathbb{R})$ as the composition of smooth functions.

(ii) Conversely, if $\varphi : (X_1, \mathcal{F}_{X_1}, \mathcal{C}_{X_1}) \times \cdots \times (X_n, \mathcal{F}_{X_n}, \mathcal{C}_{X_n}) \to (X, \mathcal{F}_X, \mathcal{C}_X)$, where $X_1, \dots, X_n, X \in \{\mathbb{R}, \mathbb{R}_{>0}\}$, is a morphism then $\varphi : X_1 \times \cdots \times X_n \to X$ is differentiable. To see this, let $x_1 \in X_1, \dots, x_n \in X_n$ and $1 \leq i \leq n$. Define $c_i(r) \coloneqq x_i + r$ and $c_j(r) \coloneqq x_j$ for $j \neq i$. Then $\varphi \circ \langle c_1, \dots, c_n \rangle \in \mathcal{C}_X$ and the partial derivative w.r.t. the i-th argument at $(x_1, \dots, x_n)$ is smooth. Since this holds for all $x_1 \in X_1, \dots, x_n \in X_n$ and $1 \leq i \leq n$, $\varphi : X_1 \times \cdots \times X_n \to X$ is differentiable.

5.1.2 Smoothed Interpretation

We interpret our language (smoothly[1]) in the cartesian closed category **Fr**. Types are interpreted as follows:

$$\begin{aligned}[\![R]\!] &\coloneqq (\mathbb{R}, C^\infty(\mathbb{R}, \mathbb{R}), C^\infty(\mathbb{R}, \mathbb{R})) \\ [\![R_{>0}]\!] &\coloneqq (\mathbb{R}_{>0}, C^\infty(\mathbb{R}, \mathbb{R}_{>0}), C^\infty(\mathbb{R}_{>0}, \mathbb{R})) \\ [\![[\mathcal{D}_1, \dots, \mathcal{D}_n]]\!] &\coloneqq (\mathbb{R}, C^\infty(\mathbb{R}, \mathbb{R}), C^\infty(\mathbb{R}, \mathbb{R}))^n \\ [\![\tau_1 \bullet \Sigma \to \tau_2]\!] &\coloneqq [\![\tau_1]\!] \times [\![\Sigma]\!] \Rightarrow [\![\tau_2]\!]\end{aligned}$$

[1] This will ensure that gradients taken in SGD with this smoothed interpretation are well-defined.

We call $[\![M]\!]_\eta$, which is defined in Fig. 5.2, the η-***smoothing*** of $[\![M]\!]$ (or of M, by abuse of language). The interpretation is mostly standard and follows Section 4.2, except for the case for conditionals in which we take a sigmoid-weighted convex combination of the branches. For primitives and conditionals we exploit Assumptions 5.1 and 5.2 and Remark 5.5(i) and the embedding morphism $\mathrm{emb}_{\tau,\tau'} : [\![\tau]\!] \to [\![\tau']\!]$ in **Fr** is defined by (as in Section 4.2):

- $\mathrm{emb}_{\iota,\iota'}(x) = x$ if $\iota \sqsubseteq \iota'$ and $x \in [\![\iota]\!] \subseteq [\![\iota']\!]$
- $\mathrm{emb}_{\tau_1\to\tau_2,\tau_1'\to\tau_2'}(f) \coloneqq x \in [\![\tau_1']\!] \mapsto \mathrm{emb}_{\tau_2,\tau_2'}(f(\mathrm{emb}_{\tau_1',\tau_1}(x)))$

5.1.3 Transformed Smooth Semantics

As for the unsmoothed interpretation, we also use a transformed version of the smoothed semantics, $[\![(-)]\!]^t_\eta$, where only the transformed sample rule needs to be changed:

$$[\![\boldsymbol{\theta}, \Gamma \mid [\mathcal{D}] \vdash_{\mathrm{tr}} \mathbf{sample}\, \mathcal{D} \rhd \varphi_{\boldsymbol{\theta}} : R]\!]^t_\eta((\boldsymbol{\theta}, \gamma), [z]) \coloneqq z$$

As in Proposition 4.10, the transformations can be factored out globally:

Proposition 5.6. *If* $\boldsymbol{\theta} \mid [\mathcal{D}_1, \ldots, \mathcal{D}_n] \vdash_{\mathrm{tr}} M : R$ *then there exists a polynomial function* $\boldsymbol{\varphi}_{(-)} : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}^n$ *which is a strong diffeomorphism and for all* $\eta > 0$ *and* $(\boldsymbol{\theta}, \boldsymbol{s}) \in \boldsymbol{\Theta} \times \mathbb{R}^n$,

$$[\![M]\!](\boldsymbol{\theta}, \boldsymbol{s}) = [\![M]\!]^t(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \qquad [\![M]\!]_\eta(\boldsymbol{\theta}, \boldsymbol{s}) = [\![M]\!]^t_\eta(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))$$

In particular, the transformation $\boldsymbol{\varphi}_{\boldsymbol{\theta}}$ is the same across all accuracy coefficients.

The result is proven by positing a suitable (but routine) logical relation and proving a Fundamental Lemma B.1 (see Appendix B). In particular, in the case for conditionals the eager semantics is crucial (cf. Example 4.1).

5.1.4 Smoothed Optimisation Problem

Finally, we can phrase a smoothed version of our Optimisation Problem 4.12:

Problem 5.7. η-Smoothed Optimisation

Given: term-in-context, $\theta_1 : \iota_1, \cdots, \theta_m : \iota_m \mid [\mathcal{D}_1, \ldots, \mathcal{D}_n] \vdash M : R$, parameter space $\boldsymbol{\Theta} \subseteq [\![\iota_1]\!] \times \cdots [\![\iota_m]\!]$, and *accuracy* coefficient $\eta > 0$

Find: $\mathrm{argmin}_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\, \mathbb{E}_{s_1\sim\mathcal{D}_1,\ldots,s_n\sim\mathcal{D}_n}\, [[\![M]\!]_\eta(\boldsymbol{\theta}, \boldsymbol{s})]$

5.2 Correctness of SGD with Reparameterisation Gradient

In this section, we wish to apply stochastic gradient descent with the reparameterisation[2] gradient estimator on the smoothing:

$$\boldsymbol{\theta}_{k+1} \coloneqq \boldsymbol{\theta}_k - \gamma_k \cdot \nabla_{\boldsymbol{\theta}}\, [\![M]\!]_\eta(\boldsymbol{\theta}_k, \boldsymbol{s}_k) \qquad \boldsymbol{s}_k \sim \boldsymbol{\mathcal{D}} \qquad \text{(SGD')}$$

[2] Recall that the setup of Section 4.3 assumes distributions have been reparameterised.

where $\theta_1 : \iota_1, \ldots, \theta_m : \iota_m \mid [\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}] \vdash M : R$ (slightly abusing notation in the trace type). Recall that by Remark 5.5(ii), $[\![M]\!]_\eta$ is differentiable and hence the iteration is well-defined.

We hope to invoke Proposition 2.11 to obtain a correctness guarantee. However, note that the boundedness pre-conditions may fail for non-compact $\boldsymbol{\Theta}$: e.g. the objective function $\theta \mapsto \mathbb{E}_{s\sim\mathcal{N}(0,1)}[s+\theta]$ is unbounded. Therefore, we assume the following henceforth:

Assumption 5.8. $\boldsymbol{\Theta} \subseteq [\![\iota_1]\!] \times \cdots \times [\![\iota_m]\!]$ is compact and convex.

As per Proposition 2.29, it now suffices to establish: for each $\eta > 0$, all partial derivatives of $[\![M]\!]_\eta$ up to order 2 are *uniformly*[3] bounded by polynomials (incl. $[\![M]\!]_\eta$ itself). For this to possibly hold, we henceforth strengthen Assumptions 5.1, 5.2 and 4.22:

Assumption 5.9. $\mathrm{Op} = \mathrm{Op}_{\mathrm{pb}} \cup \{\exp, \log, (-)^{-1}\}$, where $\mathrm{Op}_{\mathrm{pb}}$ is a set of smooth functions bounded by polynomials, partial derivatives of which up to order 2 are bounded by polynomials.

Assumption 5.10. $\sigma_\eta : \mathbb{R} \to [0,1]$ is smooth and for every $\eta > 0$,

$$\sup_{x\in\mathbb{R}} |\sigma'_\eta(x)| < \infty \qquad\qquad \sup_{x\in\mathbb{R}} |\sigma''_\eta(x)| < \infty$$

Note that we do *not* assume that there is an upper bound *uniform* in $\eta > 0$ for $\sigma'_\eta(x)$.

Example 5.11. For the logistic sigmoid (Example 5.3),

$$\sigma'_\eta(x) = \eta^{-1} \cdot \sigma_\eta(x) \cdot \sigma_\eta(-x) \qquad \sigma''_\eta(x) = \eta^{-2} \cdot \sigma_\eta(x) \cdot \sigma_\eta(-x) \cdot (1 - 2\sigma_\eta(x))$$

Therefore, Assumption 5.10 is satisfied. However, $\lim_{\eta\searrow 0} \sigma'_\eta(0) = \lim_{\eta\searrow 0} \frac{1}{4\eta} = \infty$.

Since our type systems intuitively ensure that the value function corresponding to all possible execution branches is bounded by polynomials and the smoothed semantics takes weighted averages of branches we obtain:

Lemma 5.12. *If* $\theta_1 : \iota_1^{(\mathbf{c})}, \ldots, \theta_m : \iota_m^{(\mathbf{c})} \mid \Sigma \vdash_{\mathrm{int}} M : R^{(\mathbf{p})}$ *then partial derivatives of* $[\![M]\!]^t_\eta$ *up to order 2 (incl.* $[\![M]\!]^t_\eta$ *itself) are uniformly bounded by polynomials.*

Remark 5.13. Since by Assumption 5.10, $|\sigma_\eta| \leq 1$, Lemma 5.12 can be strengthened to state that there exists a polynomial $p : \mathbb{R}^n \to \mathbb{R}$ such that *for all* $\eta > 0$ and for all $(\boldsymbol{\theta}, \boldsymbol{s}) \in \boldsymbol{\Theta} \times \mathbb{R}^n$, $|[\![M]\!]^t_\eta(\boldsymbol{\theta}, \boldsymbol{s})| \leq p(\boldsymbol{s})$. However, for partial derivatives the polynomial bound (uniform in $\boldsymbol{\theta}$) needs to depend on $\eta > 0$ since Assumption 5.10 only provides bounds for the derivatives of σ_η for fixed $\eta > 0$.

With the chain rule, Proposition 5.6 and Lemma 2.26 we obtain:

Corollary 5.14. *If* $\theta_1 : \iota_1^{(\mathbf{c})}, \ldots, \theta_m : \iota_m^{(\mathbf{c})} \mid \Sigma \vdash_{\mathrm{int}} M : R^{(\mathbf{p})}$ *then partial derivatives of* $[\![M]\!]_\eta$ *up to order 2 (incl.* $[\![M]\!]_\eta$ *itself) are uniformly bounded by polynomials.*

Finally, using Propositions 2.11, 2.28 and 2.29 we conclude:

[3] Recall that $f : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$ is uniformly bounded by p if for all $(\boldsymbol{\theta}, \boldsymbol{s}) \in \boldsymbol{\Theta} \times \mathbb{R}^n$, $|f(\boldsymbol{\theta}, \boldsymbol{s})| \leq p(\boldsymbol{s})$.

Theorem 5.15 (Unbiasedness)**.** *If* $\theta_1 : \iota_1^{(\mathbf{c})}, \dots, \theta_m : \iota_m^{(\mathbf{c})} \mid \Sigma \vdash_{\text{int}} M : R^{(\mathbf{p})}$ *and* $\eta > 0$ *then the reparameterisation gradient estimator for the* η*-smoothing is unbiased: for all* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$

$$\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[\nabla_{\boldsymbol{\theta}} [\![M]\!]_\eta(\boldsymbol{\theta}, \boldsymbol{s})] = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[[\![M]\!]_\eta(\boldsymbol{\theta}, \boldsymbol{s})]$$

NB Unbiasedness even holds for non-compact $\boldsymbol{\Theta}$: for each $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, we can simply invoke Theorem 5.15 for some closed ball around $\boldsymbol{\theta}$.

Theorem 5.16 (Correctness of SGD for Fixed-Accuracy Smoothing)**.**
If $\theta_1 : \iota_1^{(\mathbf{c})}, \dots, \theta_m : \iota_m^{(\mathbf{c})} \mid \Sigma \vdash_{\text{int}} M : R^{(\mathbf{p})}$ *and* $\eta > 0$ *then SGD for the* η*-smoothing is correct: for the iteration* (SGD')*, almost surely* $\liminf_{i\to\infty} \|\nabla\, \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[[\![M]\!]_\eta(\boldsymbol{\theta}_i, \boldsymbol{s})]\| = 0$ *or* $\boldsymbol{\theta}_i \notin \boldsymbol{\Theta}$ *for some* $i \in \mathbb{N}$.

Lemma 5.12 can be established using a similar logical predicate as for $[\![(-)]\!]$ and proving a Fundamental Lemma analogous to Lemmas 4.20 and 4.25. Formally, we stipulate the logical predicate $\mathcal{UPB}_\tau^{(n)}$ on $\boldsymbol{\Theta} \times \mathbb{R}^n \to [\![\tau]\!]$ in **Fr**:

(i) $f \in \mathcal{UPB}^{(n)}_{\iota^{(\mathbf{c})}}$ if there exists smooth $g : \boldsymbol{\Theta} \to \mathbb{R}$ satisfying $f(\boldsymbol{\theta}, \boldsymbol{s}) = g(\boldsymbol{\theta})$ for all $(\boldsymbol{\theta}, \boldsymbol{s}) \in \boldsymbol{\Theta} \times \mathbb{R}^n$

(ii) $f \in \mathcal{UPB}^{(n)}_{\iota^{(\mathbf{p})}}$ if partial derivatives of f up to order 2 (incl. f itself) are uniformly bounded by a polynomial

(iii) $f \in \mathcal{UPB}^{(n)}_{R^{(\mathbf{l})}_{>0}}$ if partial derivatives of $\log \circ f$ up to order 2 (incl. $\log \circ f$ itself) are uniformly bounded by a polynomial

(iv) $f \in \mathcal{UPB}^{(n)}_{\tau_1 \bullet \Sigma_3 \to \tau_2}$ if for all $n_2 \in \mathbb{N}$ and $g \in \mathcal{UPB}^{(n+n_2)}_{\tau_1}$, $f \odot g \in \mathcal{UPB}^{(n+n_2+|\Sigma_3|)}_{\tau_2}$

where for $f : \boldsymbol{\Theta} \times \mathbb{R}^{n_1} \to [\![\tau_1 \bullet \Sigma_3 \to \tau_2]\!]$ and $g : \boldsymbol{\Theta} \times \mathbb{R}^{n_1+n_2} \to [\![\tau_1]\!]$ we define

$$\begin{aligned} f \odot g : \boldsymbol{\Theta} \times \mathbb{R}^{n_1+n_2+|\Sigma_3|} &\to \tau_2 \\ (\boldsymbol{\theta}, \boldsymbol{s}_1 +\!\!+ \boldsymbol{s}_2 +\!\!+ \boldsymbol{s}_3) &\mapsto f(\boldsymbol{\theta}, \boldsymbol{s}_1)(g(\boldsymbol{\theta}, \boldsymbol{s}_1 +\!\!+ \boldsymbol{s}_2), \boldsymbol{s}_3) \end{aligned}$$

In contrast to Section 4.5, where we were just concerned with integrability, we cannot fix the parameters here, and we need to establish *uniform* bounds.

Lemma 5.17. *If* $f \in \mathcal{UPB}^{(n)}_\tau$ *and* $\tau \sqsubseteq_{\text{int}} \tau'$ *then*[4] $\text{emb}_{\tau,\tau'}(f) \in \mathcal{UPB}^{(n)}_{\tau'}$.

Proof. If $f \in \mathcal{UPB}^{(n)}_{\iota_1^{(\mathbf{c})}}$ then there exists smooth $g : \boldsymbol{\Theta} \to \mathbb{R}$ such that $f(\boldsymbol{\theta}, \boldsymbol{s}) = g(\boldsymbol{\theta})$. Hence, since $\boldsymbol{\Theta}$ is compact, f is bounded by the constant $\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} |g(\boldsymbol{\theta})|$ (and similarly for the partial derivatives). Consequently,

$$\text{emb}_{\iota_1^{(\mathbf{c})}, \iota_2^{(\mathbf{p})}}(f) \in \mathcal{UPB}^{(n)}_{\iota_2^{(\mathbf{p})}}$$

provided $\iota_1 \sqsubseteq \iota_2$. Analogously, if $f \in \mathcal{UPB}^{(n)}_{R^{(\mathbf{c})}_{>0}}$ and $f(\boldsymbol{\theta}, \boldsymbol{s}) = g(\boldsymbol{\theta})$ for smooth $g : \boldsymbol{\Theta} \to \mathbb{R}$,

$$c \coloneqq \inf_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} g(\boldsymbol{\theta}) > -\infty \qquad\qquad d \coloneqq \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} g(\boldsymbol{\theta}) < \infty$$

[4] where emb is defined as in Section 4.2 ignoring the annotation

(by compactness of $\boldsymbol{\Theta}$) and $\log \circ f$ is bounded by $\max\{|\log c|, |\log d|\}$ and

$$\mathrm{emb}_{\iota_1^{(\mathbf{c})},\iota_2^{(\mathbf{p})}}(f) \in \mathcal{UPB}^{(n)}_{R^{(\mathbf{l})}_{>0}}$$

The remaining cases are straightforward. □

We can then prove a fundamental lemma similar to Lemmas 4.20 and 4.25:

Lemma 5.18 (Fundamental)**.** *If* $\theta_1 : \iota_1^{(\mathbf{c})}, \dots, \theta_m : \iota_m^{(\mathbf{c})}, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma \vdash_{\mathrm{int}} M : \tau$, $n \in \mathbb{N}$ *and* $\xi_1 \in \mathcal{UPB}^{(n)}_{\tau_1}, \dots, \xi_\ell \in \mathcal{UPB}^{(n)}_{\tau_\ell}$ *then* $[\![M]\!] * \langle \xi_1, \dots, \xi_\ell \rangle \in \mathcal{UPB}^{(n+|\Sigma|)}_{\tau}$*, where*

$$[\![M]\!]^t_\eta * \langle \xi_1, \dots, \xi_\ell \rangle(\boldsymbol{\theta}, \boldsymbol{s} +\!\!+ \boldsymbol{s}') \coloneqq [\![M]\!](\boldsymbol{\theta}, (\xi_1(\boldsymbol{s}), \dots, \xi_\ell(\boldsymbol{s})), \boldsymbol{s}')$$

Proof. We prove the claim by induction on the type derivation of M.

- For subtyping we exploit Lemma 5.17.
- For variables (incl. θ_i) this is immediate.
- For $\boldsymbol{\theta}, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma_1 +\!\!+ \dots +\!\!+ \Sigma_\ell \vdash_{\mathrm{tr}} \underline{f}(M_1, \dots, M_\ell) : \iota^{(\mathbf{p})}$, where $f \in \mathrm{Op}_{\mathrm{pb}}$, we exploit the chain rule, the inductive hypothesis, Assumption 5.9 and Lemma 2.27(i).
- For $\boldsymbol{\theta}, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma_1 +\!\!+ \dots +\!\!+ \Sigma_\ell \vdash_{\mathrm{tr}} \underline{f}(M_1, \dots, M_\ell) : \iota^{(\mathbf{c})}$, where $f \in \mathrm{Op}$, we exploit Assumption 5.9 and the fact that smooth functions are closed under composition.
- For the additional rules for multiplication, inverses, logarithms and exponential functions we exploit the respective elementary logarithm rule and the inductive hypothesis.
- For conditionals we exploit the chain rule, the inductive hypothesis, Assumption 5.10 and Lemma 2.27(i).
- Abstractions and appplications are covered as in Lemma 4.20.
- Finally, $[\![\mathbf{sample}\, \mathcal{D} \rhd \varphi_{\boldsymbol{\theta}}]\!] * \langle \xi_1, \dots, \xi_\ell \rangle(\boldsymbol{\theta}, \boldsymbol{z} +\!\!+ [z']) = z'$. □

5.3 Uniform Convergence

In the preceding section we have shown that stochastic gradient descent with the reparameterisation gradient can be employed to correctly (in the sense of Proposition 2.11) solve the Smoothed Optimisation Problem 5.7 for any fixed accuracy coefficient. However, *a priori*, it is not clear to what extent a solution of the Smoothed Problem 5.7 can help to solve the original Problem 4.12.

Our prime example for σ_η, the logistic sigmoid function, converges (pointwisely) to the Heaviside function on $\mathbb{R} \setminus \{0\}$ (cf. Fig. 5.1a). However, the following illustrates the potential for significant discrepancies:

Example 5.19. (i) $\mathbb{E}_{z\sim\mathcal{N}(\theta,1)}[[\![\mathbf{if}\, 0 < 0\, \mathbf{then}\, 0\, \mathbf{else}\, 1]\!]_\eta(z)] = \frac{1}{2}$ converges *nowhere* to $\mathbb{E}_{z\sim\mathcal{N}(\theta,1)}[[\![\mathbf{if}\, 0 < 0\, \mathbf{then}\, 0\, \mathbf{else}\, 1]\!](z)] = 1$.

(ii) Consider

$$M \equiv \mathbf{if}\, 0 < 0\, \mathbf{then}\, \theta \cdot \theta + 1\, \mathbf{else}\, (\theta - 1) \cdot (\theta - 1)$$

Notice that the global minimum and the only stationary point of $[\![M]\!]_\eta$ is at $\theta = \frac{1}{2}$ regardless of $\eta > 0$, where $[\![M]\!]_\eta(\frac{1}{2}) = \frac{3}{4}$. On the other hand $[\![M]\!](\frac{1}{2}) = \frac{1}{4}$ and the global minimum of $[\![M]\!]$ is at $\theta = 1$ (see Fig. 5.3).

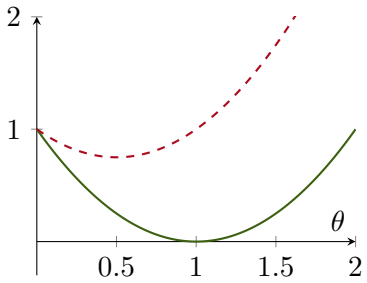


Figure 5.3: $[\![M]\!]$ (solid, green) and $[\![M]\!]_\eta$ (dashed, red) for Example 5.19(ii).

In this section we investigate under which conditions the smoothed objective function converges to the original objective function *uniformly* in $\boldsymbol{\theta} \in \boldsymbol{\Theta}$,

$$\mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}}\left[[\![M]\!]_\eta(\boldsymbol{\theta},\boldsymbol{s})\right] \xrightarrow{\text{unif.}} \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}}\left[[\![M]\!](\boldsymbol{\theta},\boldsymbol{s})\right] \qquad \text{as } \eta \searrow 0 \text{ for } \boldsymbol{\theta} \in \boldsymbol{\Theta}$$

and we design a type system guaranteeing this.

The significance of uniform convergence is that *before* running SGD, for every error tolerance $\epsilon > 0$ an accuracy coefficient $\eta > 0$ can be selected such that the difference between the smoothed and original objective function does not exceed ϵ. In particular this holds for $\boldsymbol{\theta}^*$ delivered by the SGD run for the η-smoothed problem.

For this to hold we clearly need to require that σ_η has good (uniform) convergence properties (as far as the unavoidable discontinuity at 0 allows for):

Assumption 5.20. $\sigma_\eta : \mathbb{R} \to [0,1]$ is smooth and for all $\delta > 0$, $\sigma_\eta \to [(-) \geq 0]$ uniformly on $(-\infty, -\delta) \cup (\delta, \infty)$.

This is for instance satisfied by the logistic sigmoid (Example 5.3).

5.3.1 Warm-Up: Simple Function Calculus

As a first step, we investigate the simple function calculus of Section 4.4. To rule out the pathologies of Example 5.19 and show that $[\![F]\!]_\eta$ to $[\![F]\!]$ at least *almost everywhere*, we require that guards of conditionals are almost everywhere not 0. We call such guards *safe*. This is motivated by:

Lemma 5.21. (i) *If* $f_\eta \xrightarrow{\text{a.e.}} f$ *and* $f \neq 0$ *almost everywhere then* $\sigma_\eta \circ f_\eta \xrightarrow{\text{a.e.}} [f(-) \geq 0]$.
(ii) *If* $f_\eta \xrightarrow{\text{a.e.}} f$, $g_\eta \xrightarrow{\text{a.e.}} g$ *and* $h_\eta \xrightarrow{\text{a.e.}} h$ *and* $f \neq 0$ *almost everywhere then*

$$(\sigma_\eta \circ (-f_\eta)) \cdot g + (\sigma_\eta \circ f_\eta) \cdot h \xrightarrow{\text{a.e.}} [f(-) < 0] \cdot g + [f(-) \geq 0] \cdot h$$

Proof. The second part is an easy consequence of the first.

Hence, suppose there exists a negligible set $U \subseteq \mathbb{R}^n$ such that for all $\boldsymbol{z} \in \mathbb{R}^n \setminus U$, $f_\eta(\boldsymbol{z}) \to f(\boldsymbol{z}) \neq 0$.

Let $\boldsymbol{z} \in \mathbb{R}^n \setminus U$ and $\epsilon > 0$. We assume $f(\boldsymbol{z}) < 0$ (otherwise the reasoning is similar). Due to $f_\eta(\boldsymbol{z}) \to f(\boldsymbol{z})$ and $f(\boldsymbol{z}) < 0$ there exists $\delta, \eta_0 > 0$ satisfying $f_\eta(\boldsymbol{z}) < -\delta$ for all $0 < \eta < \eta_0$. By uniform convergence of $\sigma_\eta \xrightarrow{\text{unif.}} [(-) \geq 0]$ on $(-\infty, -\delta)$ (Assumption 5.20) there exists $0 < \eta_1 < \eta_0$ satisfying $\sigma_\eta(z) < \epsilon$ for all $0 < \eta < \eta_1$ and $z < -\delta$. In particular

$$|\sigma_\eta(\underbrace{f_\eta(\boldsymbol{z})}_{<-\delta}) - \underbrace{[f(\boldsymbol{z}) \geq 0]}_{=0}| < \epsilon \qquad \square$$

We define terms $\Lambda^{\mathrm{SFC}}_{\neq 0}$ which are almost everywhere not 0 by only allowing primitives which are almost everywhere not 0 and by ruling out applying primitives to complex terms:

$$\frac{}{\underline{f}(z_{i_1},\dots,z_{i_k}) \in \Lambda^{\mathrm{SFC}}_{\neq 0}}\ f \neq 0 \text{ a.e.} \qquad \frac{G_1 \in \Lambda^{\mathrm{SFC}}_{\neq 0} \quad G_2 \in \Lambda^{\mathrm{SFC}}_{\neq 0} \quad G_3 \in \Lambda^{\mathrm{SFC}}_{\neq 0}}{\mathbf{if}\, G_1 < 0\, \mathbf{then}\, G_2\, \mathbf{else}\, G_3 \in \Lambda^{\mathrm{SFC}}_{\neq 0}}$$

where the variables $z_{i_1},\dots,z_{i_k}$ are pairwise distinct. This restriction is important because for instance subtraction $- : \mathbb{R}^2 \to \mathbb{R}$ is not 0 almost everywhere but $z_i \mapsto z_i - z_i$ is constantly 0. By Lemma 2.31, the requirement $f \neq 0$ a.e. can be guaranteed if f is analytic and not constantly 0.

Employing Lemma 5.21(ii), a simple induction shows that for $G \in \Lambda^{\mathrm{SFC}}_{\neq 0}$, $[\![G]\!] \neq 0$ almost everywhere and $[\![G]\!]_\eta \xrightarrow{\text{a.e.}} [\![G]\!]$.

Next, we can define *guard safe* simple function calculus expressions $\Lambda^{\mathrm{SFC}}_{\mathrm{s}}$ by restricting the guards:

$$\frac{}{z_j \in \Lambda^{\mathrm{SFC}}_{\mathrm{s}}} \qquad \frac{S_1 \in \Lambda^{\mathrm{SFC}}_{\mathrm{s}} \quad \cdots \quad S_\ell \in \Lambda^{\mathrm{SFC}}_{\mathrm{s}}}{\underline{f}(S_1,\dots,S_\ell) \in \Lambda^{\mathrm{SFC}}_{\mathrm{s}}} \qquad \frac{G \in \Lambda^{\mathrm{SFC}}_{\neq 0} \quad S_1 \in \Lambda^{\mathrm{SFC}}_{\mathrm{s}} \quad S_2 \in \Lambda^{\mathrm{SFC}}_{\mathrm{s}}}{\mathbf{if}\, G < 0\, \mathbf{then}\, S_1\, \mathbf{else}\, S_2 \in \Lambda^{\mathrm{SFC}}_{\mathrm{s}}}$$

A similar induction demonstrates $[\![S]\!]_\eta \xrightarrow{\text{a.e.}} [\![S]\!]$ for $S \in \Lambda^{\mathrm{SFC}}_{\mathrm{s}}$.

Uniform Convergence of Expectations

Next, we would like to not only show the almost everywhere convergence of $[\![S]\!]_\eta$ to $[\![S]\!]$, but also to prove that their *expectations* converge *uniformly* (for *all* parameters). We obtain this result (Corollary 5.23) as a consequence of the following (and Remark 5.13):

Lemma 5.22. *Suppose f_η is a sequence of functions $\mathbb{R}^n \to \mathbb{R}$ (for $\eta > 0$) and $f : \mathbb{R}^n \to \mathbb{R}$ satisfying*

(i) $f_\eta \xrightarrow{\text{a.e.}} f$
(ii) *f_η and f are bounded by a polynomial $p : \mathbb{R}^n \to \mathbb{R}$, i.e. for all $\boldsymbol{z} \in \mathbb{R}^n$ and $\eta > 0$, $|f_\eta(\boldsymbol{z})|, |f(\boldsymbol{z})| \leq p(\boldsymbol{z})$.*

Then

$$\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[f_\eta(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))\right] \xrightarrow{\text{unif.}} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[f(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))\right] \qquad \text{as } \eta \searrow 0 \text{ for } \boldsymbol{\theta} \in \boldsymbol{\Theta}$$

Corollary 5.23. *If $S \in \Lambda^{\mathrm{SFC}}_{\mathrm{s}}$ then*

$$\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[[\![F]\!](\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))] \xrightarrow{\text{unif.}} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[[\![F]\!]_\eta(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))] \qquad \text{as } \eta \searrow 0 \text{ for } \boldsymbol{\theta} \in \boldsymbol{\Theta}$$

Proof of Lemma 5.22. Intuitively, if $f_\eta \to f$ *almost everywhere* then the set where f_η and f differ by any amount can be made arbitrarily small.

Formally, to show uniform convergence, let $\epsilon > 0$. We define

$$U_k \coloneqq \left\{ \boldsymbol{z} \in \operatorname{supp}(\boldsymbol{\mathcal{D}}) \mid |f_\eta(\boldsymbol{z}) - f(\boldsymbol{z})| > \frac{\epsilon}{2} \text{ for some } 0 < \eta < \frac{1}{k} \right\}$$

Note that $(U_k)_{k\in\mathbb{N}}$ is a non-increasing sequence of sets (i.e. $U_1 \supseteq U_2 \supseteq \ldots$) and

$$\bigcap_{k\in\mathbb{N}} U_k \subseteq \{\boldsymbol{z} \in \mathbb{R}^n \mid f_\eta(\boldsymbol{z}) \not\to f(\boldsymbol{z})\}$$

which is (Lebesgue-)negligible by almost everywhere convergence. By Lemma 5.24 below,

$$\mu(U) \coloneqq \int_U p(\boldsymbol{z}) \cdot \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z}$$

where $\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) \coloneqq \boldsymbol{\mathcal{D}}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))\cdot|\det \mathbf{J}\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})|$, is a *finite* measure and absolutely continuous w.r.t. the Lebesgue measure. Thus, by continuity from above (of μ) there exists k such that $\mu(U_k) < \frac{\epsilon}{4}$. Finally, it suffices to observe that for $0 < \eta < \frac{1}{k}$ and $\boldsymbol{\theta} \in \boldsymbol{\Theta}$:

$$\begin{aligned}
&|\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[f_\eta(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))] - \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[f(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))]| \\
&\leq \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}[[\boldsymbol{z} \in U_k] \cdot |f_\eta(\boldsymbol{z}) - f(\boldsymbol{z})|] + \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}[[\boldsymbol{z} \notin U_k] \cdot |f_\eta(\boldsymbol{z}) - f(\boldsymbol{z})|] \\
&\leq \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}[[\boldsymbol{z} \in U_k] \cdot 2 \cdot p(\boldsymbol{z})] + \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}\left[[\boldsymbol{z} \notin U_k] \cdot \frac{\epsilon}{2}\right] \\
&\leq 2 \cdot \mu(U_k) + \frac{\epsilon}{2} \leq \epsilon
\end{aligned}$$

□

Lemma 5.24. *If $f : U \to \mathbb{R}$, where $U \subseteq \mathbb{R}^n$, is a Schwartz function, $\boldsymbol{\varphi}_{(-)} : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}^n$ is a polynomial and a strong diffeomorphism, and $p : \mathbb{R}^n \to \mathbb{R}$ is a polynomial then*

$$\int_U \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left|f(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) \cdot \det \mathbf{J}\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}) \cdot p(\boldsymbol{z})\right| \mathrm{d}\boldsymbol{z} < \infty$$

Proof. Note that $\det \mathbf{J}\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s})$ is a polynomial $p_1 : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$. Since each $\boldsymbol{\varphi}_{\boldsymbol{\theta}}$ is a diffeomorphism and p is continuous, p is either positive or negative. W.l.o.g. we assume the former. Thus,

$$|\det \mathbf{J}\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))| = \frac{1}{|\det \mathbf{J}\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s})|} = \frac{1}{p_1(\boldsymbol{\theta}, \boldsymbol{s})}$$

Since $\boldsymbol{\varphi}_{(-)}$ is assumed to be a *strong* diffeomorphism (see Definition 4.2(ii)), there exists $c > 0$ such that $p_1(\boldsymbol{\theta}, \boldsymbol{s}) > c$ and by Lemma 2.26 there exists a polynomial $p_2 : \mathbb{R}^n \to \mathbb{R}$ satisfying

$$\left|\|\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s})\|^{n+3} + p(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))\right| \leq p_2(\boldsymbol{s})$$

for all $(\boldsymbol{\theta}, \boldsymbol{s}) \in \boldsymbol{\Theta} \times U$. Hence,

$$\begin{aligned}
&\sup_{(\boldsymbol{\theta},\boldsymbol{z})\in\boldsymbol{\Theta}\times U} \|\boldsymbol{z}\|^{n+3} \cdot \left|f(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) \cdot \det \mathbf{J}\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}) \cdot p(\boldsymbol{z})\right| \\
&= \sup_{(\boldsymbol{\theta},\boldsymbol{s})\in\boldsymbol{\Theta}\times U} \|\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s})\|^{n+3} \cdot |f(\boldsymbol{s})| \cdot \underbrace{\left|\frac{1}{p_1(\boldsymbol{\theta}, \boldsymbol{s})}\right|}_{\leq 1/c} \cdot |p(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))| \\
&= \frac{1}{c} \cdot \sup_{\boldsymbol{s}\in U} p_2(\boldsymbol{s}) \cdot |f(\boldsymbol{s})| < \infty
\end{aligned}$$

by definition of Schwartz functions and the claim follows with Lemma 2.24. □

5.3.2 Extension to the Full Language

Next, we wish to design a type system $\vdash_{\text{unif}}$ for our full language of Chapter 4 and generalise the uniform convergence result. In particular, we would like to support guards with abstractions, higher-order features, nestings of primitive operations and parameters. As in the previous subsection, we will need to ensure that guards are not 0 almost everywhere.

As a consequence of the uniform limit theorem [Munkres, 1999], uniform convergence can only possibly hold if the *expectation* $\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[[\![M]\!](\boldsymbol{\theta},\boldsymbol{s})]$ is continuous (as a function of the parameters $\boldsymbol{\theta}$).

Example 5.25. (i) For a straightforward counterexample take

$$M \equiv \mathbf{if}\,\theta < 0\,\mathbf{then}\,0\,\mathbf{else}\,1$$

We have $\mathbb{E}_{\boldsymbol{s}}[[\![M]\!](\theta)] = [\theta \geq 0]$ which is discontinuous at $\theta = 0$.

(ii) On the other hand $M \equiv \mathbf{if}\,(\mathbf{sample}_{\mathcal{N}} \rhd \varphi_\theta) < 0\,\mathbf{then}\,0\,\mathbf{else}\,1$ is no problem because

$$\begin{aligned}\mathbb{E}_{s\sim\mathcal{N}}[[\![M]\!]_\eta(s)] &= \mathbb{E}_{s\sim\mathcal{N}}[[\![\mathbf{if}\,z<0\,\mathbf{then}\,0\,\mathbf{else}\,1]\!]_\eta(\varphi_\theta(s))] \\ &\xrightarrow{\text{unif.}} \mathbb{E}_{s\sim\mathcal{N}}[\underbrace{[\![\mathbf{if}\,z<0\,\mathbf{then}\,0\,\mathbf{else}\,1]\!]}_{\in\Lambda^{\text{SFC}}}(\varphi_\theta(s))] = \mathbb{E}_{s\sim\mathcal{N}}[[\![M]\!](s)]\end{aligned}$$

as we have seen above.

Consequently, we require that guards do not depend directly on parameters, but they may do so, indirectly, via a diffeomorphic reparameterisation transform; see Example 4.5.

We extend our approach described in the previous subsection: intuitively, we ensure that guards are the composition of a diffeomorphic transformation of the random samples (potentially depending on parameters) and a function which does not vanish almost everywhere. We call such guards *safe*. In particular, safe guards may only depend on parameters indirectly via a diffeomorphic reparameterisation.

Motivated by the dichotomy property Lemma 2.31 of analytic functions we henceforth assume:

Assumption 5.26. Primitive operations in Op are analytic.

(In this section we do not require Assumption 5.9, apart from in Theorem 5.36, where it can be used to guarantee integrability.)

Type System for Uniform Convergence

Allowing more complex guards than simple function calculus terms in $\Lambda^{\text{SFC}}_{\neq 0}$ can render spotting safe guards tricky:

Example 5.27. Consider the following terms (omitting irrelevant transformations):

$$\begin{aligned}M_1 &\equiv (\lambda x.\,\mathbf{if}\,x - x < 0\,\mathbf{then}\,0\,\mathbf{else}\,1)\,\mathbf{sample}_{\mathcal{N}} \\ M_2 &\equiv \mathbf{if}\,(\mathbf{sample}_{\mathcal{N}} - \mathbf{sample}_{\mathcal{N}}) < 0\,\mathbf{then}\,M\,\mathbf{else}\,N \\ M_3 &\equiv (\lambda y, z.\,\mathbf{if}\,y - z < 0\,\mathbf{then}\,0\,\mathbf{else}\,1)\,\mathbf{sample}_{\mathcal{N}}\,\mathbf{sample}_{\mathcal{N}}\end{aligned}$$

$$M_4 \equiv (\lambda x.\,(\lambda y, z.\,\mathbf{if}\; y - z < 0\,\mathbf{then}\,0\,\mathbf{else}\,1)\,x\,x)\,\mathbf{sample}_{\mathcal{N}}$$
$$M_5 \equiv (\lambda f.\,\mathbf{if}\;(f\,\mathbf{sample}_{\mathcal{N}})^2 < 0\,\mathbf{then}\,0\,\mathbf{else}\,1)\,(\lambda x.\,x - \mathbf{sample}_{\mathcal{N}})$$

Uniform convergence and guard safety only fail for M_1 and M_4. Intuitively, when evaluated, the terms $x - x$ in the guard of M_1 and $y - z$ in the guard of M_4 have denotation 0. This is not the case for M_3 because y and z are replaced with different samples.

In particular, the difference between M_3 and M_4 suggests that we need to keep track of dependencies on samples to detect unsafe guards.

Hence, we propose another instance of the generic type system of Section 4.6, $\vdash_{\text{unif}}$, where we annotate base types by (σ, dep), where $\sigma \in \{\mathbf{f}, \mathbf{t}\}$ denotes whether we seek to establish guard safety and dep is a finite set of identifiers α_j capturing possible dependencies on samples.

trace types	$\Sigma ::= [\mathcal{D}_1, \ldots, \mathcal{D}_n]$
base types	$\iota ::= R \mid R_{>0}$
types	$\tau ::= R^{(\sigma,\text{dep})} \mid R_{>0}^{(\sigma,\text{dep})} \mid \tau \bullet \Sigma \to \tau$

The subtyping relation is defined in Fig. 5.4a. Intuitively, we can always drop[5] guard safety and add more dependencies.

For conditionals we require that only safe guards are used:

$$\frac{\Gamma \mid \Sigma \vdash_{\text{unif}} L : R^{(\mathbf{t},\text{dep}')} \quad \Gamma \mid \Sigma' \vdash_{\text{unif}} M : \iota^{(\sigma,\text{dep})} \quad \Gamma \mid \Sigma'' \vdash_{\text{unif}} N : \iota^{(\sigma,\text{dep})}}{\Gamma \mid \Sigma +\!\!+ \Sigma' +\!\!+ \Sigma'' \vdash_{\text{unif}} \mathbf{if}\; L < 0\,\mathbf{then}\,M\,\mathbf{else}\,N : \iota^{(\sigma,\text{dep})}}$$

Perhaps surprisingly, we only include the dependencies of the branches and not the guards. (This is sufficient to show that if σ is *true* then on each piece, denotations are almost everywhere not 0; see below.)

Samples receive an identifier and their transformation is safe to be used in guards:

$$\frac{}{\boldsymbol{\theta}, \Gamma \mid [\mathcal{D}] \vdash_{\text{unif}} \mathbf{sample}_{\mathcal{D}} \rhd \varphi_{\boldsymbol{\theta}} : R^{(\mathbf{t},\{\alpha_j\})}}\; \varphi_{(-)} \in \text{Diff}_{\boldsymbol{\Theta}}(\text{supp}(\mathcal{D}))$$

Remark 5.28. We do not strictly require that α_j is a *fresh* identifier. In particular, $\mathbf{sample}_{\mathcal{N}} - \mathbf{sample}_{\mathcal{N}}$ (omitting irrelevant transformations) can be typed (cf. Fig. 5.4b) as both $\mid [\mathcal{N}, \mathcal{N}] \vdash_{\text{unif}} \mathbf{sample}_{\mathcal{N}} - \mathbf{sample}_{\mathcal{N}} : R^{(\mathbf{t},\{\alpha_1,\alpha_2\})}$ and (using the same identifier for both samples) $\mid [\mathcal{N}, \mathcal{N}] \vdash_{\text{unif}} \mathbf{sample}_{\mathcal{N}} - \mathbf{sample}_{\mathcal{N}} : R^{(\mathbf{f},\{\alpha_1\})}$. However, in the second regime the guard safety flag is necessarily $\mathbf{f}$ (cf. the next rule). In general, to recognise guard safe terms as such, α_j should be chosen fresh.

For primitives either the guard safety flag needs to be $\mathbf{f}$ (false) or

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\text{unif}} M_1 : \iota_1^{(\mathbf{t},\text{dep}_1)} \quad \cdots \quad \Gamma \mid \Sigma_\ell \vdash_{\text{unif}} M_\ell : \iota_\ell^{(\mathbf{t},\text{dep}_\ell)}}{\Gamma \mid \Sigma_1 +\!\!+ \cdots +\!\!+ \Sigma_\ell \vdash_{\text{unif}} \underline{f}(M_1, \ldots, M_\ell) : \iota^{(\mathbf{t},\text{dep}_1 \cup \cdots \cup \text{dep}_\ell)}}$$

[5] as long as it is not used in guards

$$\frac{}{\iota_1^{(\sigma,\mathrm{dep})} \sqsubseteq_{\mathrm{unif}} \iota_2^{(\sigma,\mathrm{dep}')}}\ \iota_1 \sqsubseteq \iota_2, \mathrm{dep} \subseteq \mathrm{dep}' \qquad \frac{}{\iota_1^{(\mathbf{t},\mathrm{dep})} \sqsubseteq_{\mathrm{unif}} \iota_2^{(\mathbf{f},\mathrm{dep}')}}\ \iota_1 \sqsubseteq \iota_2, \mathrm{dep} \subseteq \mathrm{dep}'$$

$$\frac{\tau_1' \sqsubseteq_{\mathrm{unif}} \tau_1 \quad \tau_2 \sqsubseteq_{\mathrm{unif}} \tau_2'}{(\tau_1 \bullet \Sigma \to \tau_2) \sqsubseteq_{\mathrm{unif}} (\tau_1' \bullet \Sigma \to \tau_2')}$$

(a) Suptyping

$$\frac{\Gamma \mid \Sigma \vdash_{\mathrm{unif}} M : \tau}{\Gamma \mid \Sigma \vdash_{\mathrm{unif}} M : \tau'}\ \tau \sqsubseteq_{\mathrm{unif}} \tau' \qquad \frac{}{x_1 : \tau_1, \ldots, x_m : \tau_m \mid [] \vdash_{\mathrm{unif}} x_i : \tau_i}$$

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\mathrm{unif}} M_1 : \iota_1^{(\mathbf{f},\mathrm{dep})} \quad \cdots \quad \Gamma \mid \Sigma_\ell \vdash_{\mathrm{unif}} M_\ell : \iota_\ell^{(\mathbf{f},\mathrm{dep})}}{\Gamma \mid \Sigma_1 +\!\!+ \cdots +\!\!+ \Sigma_\ell \vdash_{\mathrm{unif}} \underline{f}(M_1, \ldots, M_\ell) : \iota^{(\mathbf{f},\mathrm{dep})}}\ f : [\![\iota_1]\!] \times \cdots \times [\![\iota_\ell]\!] \to [\![\iota]\!] \in \mathrm{Op}$$

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\mathrm{unif}} M_1 : \iota_1^{(\mathbf{t},\mathrm{dep}_1)} \quad \cdots \quad \Gamma \mid \Sigma_\ell \vdash_{\mathrm{unif}} M_\ell : \iota_\ell^{(\mathbf{t},\mathrm{dep}_\ell)}}{\Gamma \mid \Sigma_1 +\!\!+ \cdots +\!\!+ \Sigma_\ell \vdash_{\mathrm{unif}} \underline{f}(M_1, \ldots, M_\ell) : \iota^{(\mathbf{t},\mathrm{dep}_1 \cup \cdots \cup \mathrm{dep}_\ell)}}\ \nabla f \neq \mathbf{0} \text{ a.e.},\ \forall i \neq j.\ \mathrm{dep}_i \cap \mathrm{dep}_j = \emptyset$$

$$\frac{\Gamma \mid \Sigma \vdash_{\mathrm{unif}} L : R^{(\mathbf{t},\mathrm{dep}')} \quad \Gamma \mid \Sigma' \vdash_{\mathrm{unif}} M : \iota^{(\sigma,\mathrm{dep})} \quad \Gamma \mid \Sigma'' \vdash_{\mathrm{unif}} N : \iota^{(\sigma,\mathrm{dep})}}{\Gamma \mid \Sigma +\!\!+ \Sigma' +\!\!+ \Sigma'' \vdash_{\mathrm{unif}} \mathbf{if}\, L < 0\, \mathbf{then}\, M\, \mathbf{else}\, N : \iota^{(\sigma,\mathrm{dep})}}$$

$$\frac{\Gamma, y : \tau_1 \mid \Sigma \vdash_{\mathrm{unif}} M : \tau_2}{\Gamma \mid [] \vdash_{\mathrm{unif}} \lambda y.\, M : \tau_1 \bullet \Sigma \to \tau_2} \qquad \frac{\Gamma \mid \Sigma_1 \vdash_{\mathrm{unif}} M : \tau_1 \bullet \Sigma_3 \to \tau_2 \quad \Gamma \mid \Sigma_2 \vdash_{\mathrm{unif}} N : \tau_1}{\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 +\!\!+ \Sigma_3 \vdash_{\mathrm{unif}} M\, N : \tau_2}$$

$$\frac{}{\boldsymbol{\theta}, \Gamma \mid [\mathcal{D}] \vdash_{\mathrm{unif}} \mathbf{sample}_{\mathcal{D}} \rhd \varphi_{\boldsymbol{\theta}} : R^{(\mathbf{t},\{\alpha_j\})}}\ \varphi_{(-)} \in \mathrm{Diff}_{\boldsymbol{\Theta}}(\mathrm{supp}(\mathcal{D}))$$

(b) Typing rules for $\vdash_{\mathrm{unif}}$.

Figure 5.4: Type system for uniform convergence.

where $\nabla f \neq \mathbf{0}$ a.e. and the arguments' dependencies are pairwise disjoint. (This is designed to admit M_3 of Example 5.27 but reject M_4, cf. Example 5.29.) In particular, $\vdash_{\mathrm{unif}} \underline{r} : R^{(\sigma,\mathrm{dep})}$ only if $\sigma = \mathbf{f}$, and the following rules are admissible:

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\mathrm{unif}} M_1 : \iota^{(\mathbf{f},\mathrm{dep})} \quad \Gamma \mid \Sigma_2 \vdash_{\mathrm{unif}} M_2 : \iota^{(\mathbf{f},\mathrm{dep})}}{\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 \vdash_{\mathrm{unif}} M_1 \circ M_2 : \iota^{(\mathbf{f},\mathrm{dep})}}\ \circ \in \{+, \cdot\}$$

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\mathrm{unif}} M_1 : \iota^{(\mathbf{t},\mathrm{dep}_1)} \quad \Gamma \mid \Sigma_2 \vdash_{\mathrm{unif}} M_2 : \iota^{(\mathbf{t},\mathrm{dep}_2)}}{\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 \vdash_{\mathrm{unif}} M_1 \circ M_2 : \iota^{(\mathbf{t},\mathrm{dep}_1 \cup \mathrm{dep}_2)}}\ \circ \in \{+, \cdot\}, \mathrm{dep}_1 \cap \mathrm{dep}_2 = \emptyset$$

$$\frac{\Gamma \mid \Sigma \vdash_{\mathrm{unif}} M : R^{(\sigma,\mathrm{dep})}}{\Gamma \mid \Sigma \vdash_{\mathrm{unif}} \underline{\exp}(M) : R_{>0}^{(\sigma,\mathrm{dep})}} \qquad \frac{\Gamma \mid \Sigma \vdash_{\mathrm{unif}} M : R_{>0}^{(\sigma,\mathrm{dep})}}{\Gamma \mid \Sigma \vdash_{\mathrm{unif}} \underline{\log}(M) : R^{(\sigma,\mathrm{dep})}}$$

The full type system is presented in Fig. 5.4b.

Example 5.29. We re-visit some of the terms of Example 5.27:

(i) $M_3 \equiv (\lambda y, z.\, \mathbf{if}\, y - z < 0\, \mathbf{then}\, 0\, \mathbf{else}\, 1)\, \mathbf{sample}_{\mathcal{N}}\, \mathbf{sample}_{\mathcal{N}}$ is typabable. In particular, we claim

$$\mid [\mathcal{N}, \mathcal{N}] \vdash_{\text{unif}} M_3 : R^{(\mathbf{f},\emptyset)}$$

By the rules for sampling, applications and abstractions it suffices to show

$$y : R^{(\mathbf{t},\{\alpha_1\})}, z : R^{(\mathbf{t},\{\alpha_2\})} \mid [] \vdash_{\text{unif}} \mathbf{if}\, y - z < 0\, \mathbf{then}\, 0\, \mathbf{else}\, 1 : R^{(\mathbf{f},\emptyset)}$$

In particular, $y : R^{(\mathbf{t},\{\alpha_1\})}, z : R^{(\mathbf{t},\{\alpha_2\})} \mid [] \vdash_{\text{unif}} y - z : R^{(\mathbf{t},\text{dep})}$ is necessary. This can be derived using the second rule for primitives because the subtraction function is only stationary at $(0,0)$, $\{\alpha_1\} \cap \{\alpha_2\} = \emptyset$ and

$$y : R^{(\mathbf{t},\{\alpha_1\})}, z : R^{(\mathbf{t},\{\alpha_2\})} \mid [] \vdash_{\text{unif}} y : R^{(\mathbf{t},\{\alpha_1\})}$$
$$y : R^{(\mathbf{t},\{\alpha_1\})}, z : R^{(\mathbf{t},\{\alpha_2\})} \mid [] \vdash_{\text{unif}} z : R^{(\mathbf{t},\{\alpha_2\})}$$

(ii) The term

$$M_4 \equiv (\lambda x.\, (\lambda y, z.\, \mathbf{if}\, y - z < 0\, \mathbf{then}\, 0\, \mathbf{else}\, 1)\, x\, x)\, \mathbf{sample}_{\mathcal{N}}$$

is not typable.

$\Gamma \mid \Sigma \vdash_{\text{unif}} \mathbf{sample}_{\mathcal{N}} : \tau$ implies that for some j, τ is $R^{(\sigma,\text{dep})}$, where $\alpha_j \in \text{dep}$.

If M_4 was typabable then

$$\Gamma, x : R^{(\sigma_1,\text{dep}_1)} \mid \Sigma' \vdash_{\text{unif}} (\lambda y, z.\, \mathbf{if}\, y - z < 0\, \mathbf{then}\, 0\, \mathbf{else}\, 1)\, x\, x : \tau'$$

where $\alpha_j \in \text{dep}_1$. This in turn implies

$$\Gamma, x : R^{(\sigma_1,\text{dep}_1)}, y : R^{(\sigma_2,\text{dep}_2)}, z : R^{(\sigma_3,\text{dep}_3)} \mid \Sigma'' \vdash_{\text{unif}} y - z : R^{(\mathbf{t},\text{dep}')}$$

where $\alpha_j \in \text{dep}_2, \text{dep}_3$. By the second rule for primitives this only holds provided that $\text{dep}_2' \cap \text{dep}_3' = \emptyset$, where

$$\Gamma, x : R^{(\sigma_1,\text{dep}_1)}, y : R^{(\sigma_2,\text{dep}_2)}, z : R^{(\sigma_3,\text{dep}_3)} \vdash_{\text{unif}} y : R^{(\mathbf{t},\text{dep}_2')}$$
$$\Gamma, x : R^{(\sigma_1,\text{dep}_1)}, y : R^{(\sigma_2,\text{dep}_2)}, z : R^{(\sigma_3,\text{dep}_3)} \vdash_{\text{unif}} z : R^{(\mathbf{t},\text{dep}_3')}$$

but by the variable (and potentially subtyping rules) it necessarily holds $\text{dep}_2 \subseteq \text{dep}_2'$ and $\text{dep}_3 \subseteq \text{dep}_3'$. This is a contradiction because $\alpha_j \in \text{dep}_2, \text{dep}_3$.

(iii) The term

$$M_5 \equiv \left(\lambda f.\, \mathbf{if}\, (f\, \mathbf{sample}_{\mathcal{N}})^2 < 0\, \mathbf{then}\, 0\, \mathbf{else}\, 1\right)\, (\lambda x.\, x - \mathbf{sample}_{\mathcal{N}})$$

is typabable. Note that

$$\mid [] \vdash_{\text{unif}} \lambda x.\, x - \mathbf{sample}_{\mathcal{N}} : R^{(\mathbf{t},\{\alpha_1\})} \bullet [\mathcal{N}] \to R^{(\mathbf{t},\{\alpha_1,\alpha_2\})}$$

because subtraction is only stationary at $(0,0)$, $\{\alpha_1\} \cap \{\alpha_2\} = \emptyset$ and

$$x : R^{(\mathbf{t},\{\alpha_1\})} \mid [] \vdash_{\text{unif}} x : R^{(\mathbf{t},\{\alpha_1\})}$$
$$x : R^{(\mathbf{t},\{\alpha_1\})} \mid [\mathcal{N}] \vdash_{\text{unif}} \mathbf{sample}_{\mathcal{N}} : R^{(\mathbf{t},\{\alpha_2\})}$$

Therefore, for typability of M_5 it suffices to observe that

$$f : R^{(\mathbf{t},\{\alpha_1\})} \bullet [\mathcal{N}] \to R^{(\mathbf{t},\{\alpha_1,\alpha_2\})} \mid [\mathcal{N}, \mathcal{N}] \vdash_{\text{unif}} (f\, \mathbf{sample}_{\mathcal{N}})^2 : R^{(\mathbf{t},\{\alpha_1,\alpha_2\})}$$

because the square function is only stationary at 0 and

$$f : R^{(\mathbf{t},\{\alpha_1\})} \bullet [\mathcal{N}] \to R^{(\mathbf{t},\{\alpha_1,\alpha_2\})} \mid [\mathcal{N}, \mathcal{N}] \vdash_{\text{unif}} f\, \mathbf{sample}_{\mathcal{N}} : R^{(\mathbf{t},\{\alpha_1,\alpha_2\})}$$

Type Soundness

Next, we would like to extend the uniform convergence result to the full language of Chapter 4 for terms typable via $\vdash_{\text{unif}}$. The full language includes the parameters not only via diffeomorphic transformations of samples, but it may also contain them explicitly in branches. We show type soundness in two steps: First, we formalise correctness of the sample-dependency annotation and obtain guard safety whenever the σ-flag is $\mathbf{t}$. Afterwards, we prove convergence (in a suitable sense) and conclude uniform convergence of the expectations.

Dependencies on Samples and Guard Safety

We wish to formalise the meaning of the dependency and guard safety annotation of base types. We accomplish this by positing a logical predicate $\mathcal{DGS}_{\tau}^{(n,\text{dict})}$ on morphisms $\boldsymbol{\Theta} \times \mathbb{R}^n \to [\![\tau]\!]$ in $\mathbf{QBS}$, where $n \in \mathbb{N}$, $\text{dict} : \{1,\dots,n\} \to \text{Dep}$ and $\text{Dep} \coloneqq \{\alpha_1, \alpha_2, \dots\}$ is the set of identifiers. Intuitively, dict indicates what abstract identifiers are used for the n samples.

(i) $f \in \mathcal{DGS}_{\iota(\mathbf{f},\text{dep})}^{(n,\text{dep})}$ is unrestricted

(ii) $f \in \mathcal{DGS}_{\iota(\mathbf{t},\text{dep})}^{(n,\text{dep})}$ if there exist a partition $U_1,\dots,U_k$ of $\mathbb{R}^n$ and analytic functions $f_1,\dots,f_k : \mathbb{R}^{|\text{dict}^{-1}(\text{dep})|} \to \mathbb{R}$ satisfying

  (a) $f(\boldsymbol{\theta}, \boldsymbol{z}) = \sum_{i=1}^{n} [\boldsymbol{z} \in U_i] \cdot f_i(\boldsymbol{\pi}_{\text{dict}^{-1}(\text{dep})}(\boldsymbol{z}))$

  (b) $\nabla f_i \neq \mathbf{0}$ a.e.

(iii) $f \in \mathcal{DGS}_{\tau_1 \bullet \Sigma \to \tau_2}^{(n,\text{dep})}$ if for all $n' \in \mathbb{N}$, $\text{dict}' : \{1,\dots,n+n'\} \to \text{Dep}$ extending $\text{dict} : \{1,\dots,n\} \to \text{Dep}$ and $g \in \mathcal{DGS}_{\tau_1}^{(n+n',\text{dict}')}$,

$$f \odot g \in \mathcal{DGS}_{\tau_2}^{(n+n'+|\Sigma|,\text{dict}'')}$$

for some $\text{dict}'' : \{1,\dots,n+n'+|\Sigma|\} \to \text{Dep}$ extending dict'.

For $I = \{i_1,\dots,i_\ell\}$ such that $1 \le i_1 < \dots < i_\ell \le n$, $\boldsymbol{\pi}_I$ is the projection to the indices in I, i.e. $\boldsymbol{\pi}_I(\boldsymbol{z}) = [z_{i_1},\dots,z_{i_\ell}]$. Note that due to the dichotomy Lemma 2.31, Item (ii)b implies $f \neq 0$ almost everywhere, which will be handy for guard safety.

Following our well-tested strategy, we then prove a fundamental lemma (resembling Lemmas 5.18, 4.20 and 4.25), *mutatis mutandis*:

Lemma 5.30 (Fundamental)**.** *If* $\theta_1 : \iota_1^{(\mathbf{f},\emptyset)}, \dots, \theta_m : \iota_m^{(\mathbf{f},\emptyset)}, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma \vdash_{\text{unif}} M : \tau$, $n \in \mathbb{N}$, $\text{dict} : \{1,\dots,n\} \to \text{Dep}$, $\xi^{(1)} \in \mathcal{DGS}_{\tau_1}^{(n,\text{dict})}, \dots, \xi^{(\ell)} \in \mathcal{DGS}_{\tau_\ell}^{(n,\text{dict})}$ *then*

$$[\![M]\!]^t * \langle \xi^{(1)}, \dots, \xi^{(\ell)} \rangle \in \mathcal{DGS}_{\tau}^{(n+|\Sigma|,\text{dict}')}$$

for some $\text{dict}' : \mathbb{R}^{n+|\Sigma|} \to \mathbb{R}$ *extending* dict*, where*

$$\left([\![M]\!]^t * \langle \xi^{(1)}, \dots, \xi^{(\ell)} \rangle \right) (\boldsymbol{\theta}, \boldsymbol{z} +\!\!+ \boldsymbol{z}') \coloneqq [\![M]\!]^t (\boldsymbol{\theta}, (\xi^{(1)}(\boldsymbol{\theta}, \boldsymbol{z}), \dots, \xi^{(\ell)}(\boldsymbol{\theta}, \boldsymbol{z})), \boldsymbol{z}')$$

Intuitively, in the case for primitives we take the intersection of the subterms' partitions and apply Lemma 2.32 (after renaming variables). Besides, for transformed samples $\boldsymbol{\theta}, \Gamma \mid [\mathcal{D}] \vdash_{\text{unif}} \mathbf{sample}_{\mathcal{D}} \rhd \varphi_{\boldsymbol{\theta}} : R^{(\mathbf{t},\{\alpha_j\})}$, dep is extended as $\text{dep}[n+1 \mapsto \alpha_j]$. Details are provided in Appendix B.2.

Uniform Convergence

We are almost ready to show uniform convergence of the expectations. However, a further complication is the fact that branches may contain parameters. To account for them, we introduce the following notion, formalising convergence uniformly in $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and *almost* uniformly in $\boldsymbol{z} \in \mathbb{R}^n$:

Definition 5.31. Let $f, f_\eta : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$. We say that f_η ***converges uniformly almost uniformly*** to f (notation: $f_\eta \xrightarrow{\text{u.a.u.}} f$) if for all bounded $U \subseteq \mathbb{R}^n$ and $\epsilon > 0$ there exists $U' \subseteq U$ such that $\mathrm{Leb}(U') < \epsilon$ and $f_\eta \xrightarrow{\text{unif.}} f$ as $\eta \searrow 0$ on $\boldsymbol{\Theta} \times (\mathbb{R}^n \setminus U')$.

Remark 5.32. Let $f_\eta, f : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$ and $g_\eta, g : \mathbb{R}^n \to \mathbb{R}$ be such that $f_\eta(\boldsymbol{\theta}, \boldsymbol{z}) = g_\eta(\boldsymbol{z})$ and $f(\boldsymbol{\theta}, \boldsymbol{z}) = g(\boldsymbol{z})$ for all $(\boldsymbol{\theta}, \boldsymbol{z}) \in \boldsymbol{\Theta} \times \mathbb{R}^n$ (i.e. f_η and f are independent of parameters). By Lemma 2.34, $f_\eta \xrightarrow{\text{u.a.u.}} f$ iff $g_\eta \xrightarrow{\text{a.e.}} g$.

Example 5.33. Suppose $\Theta \subseteq \mathbb{R}$ is compact and

$$f_\eta(\theta, z) \coloneqq \sigma_\eta(z) \cdot \theta \qquad\qquad f(\theta, z) \coloneqq [z \geq 0] \cdot \theta$$

To demonstrate that $f_\eta \xrightarrow{\text{u.a.u.}} f$, let $\epsilon > 0$. Let $U' \coloneqq (-\epsilon/2, \epsilon/2)$ for which $\mathrm{Leb}(U') \leq \epsilon$. To show that $f_\eta \xrightarrow{\text{unif.}} f$ as $\eta \searrow 0$ on $\mathbb{R} \setminus U'$, we fix $\epsilon' > 0$. Let $c \coloneqq \sup_{\theta \in \Theta} |\theta|$. By Assumption 5.20, there exists $\eta_0 > 0$ such that for all $0 < \eta < \eta_0$ and $|y| \geq \epsilon/2$, $|\sigma_\eta(y) - [y \geq 0]| \leq \epsilon'/c$. Consequently, for every $0 < \eta < \eta_0$ and $(\theta, z) \in \Theta \times (\mathbb{R} \setminus U')$,

$$|\sigma_\eta(z) \cdot \theta - [z \geq 0] \cdot \theta| = |\sigma_\eta(z) - [z \geq 0]| \cdot |\theta| \leq \epsilon'$$

(Note that in this 1-dimensional example there is no need to fix a compact U.)

In general, we show:

Proposition 5.34. *If* $\theta_1 : \iota_1^{(\mathbf{f},\emptyset)}, \ldots, \theta_m : \iota_m^{(\mathbf{f},\emptyset)} \mid \Sigma \vdash_{\text{unif}} M : R^{(\sigma,\text{dep})}$ *then*

$$[\![M]\!]_\eta^t \xrightarrow{\text{u.a.u.}} [\![M]\!]^t$$

With this we can obtain the following:

Proposition 5.35. *If* $\theta_1 : \iota_1^{(\mathbf{f},\emptyset)}, \ldots, \theta_m : \iota_m^{(\mathbf{f},\emptyset)} \mid \Sigma \vdash_{\text{unif}} M : R^{(\sigma,\text{dep})}$ *and there exists a polynomial* $p : \mathbb{R}^n \to \mathbb{R}$ *satisfying*

$$|[\![M]\!]^t(\boldsymbol{\theta}, \boldsymbol{z})| \leq p(\boldsymbol{z}) \qquad\qquad |[\![M]\!]_\eta^t(\boldsymbol{\theta}, \boldsymbol{z})| \leq p(\boldsymbol{z})$$

for all $(\boldsymbol{\theta}, \boldsymbol{z}) \in \boldsymbol{\Theta} \times \mathbb{R}^n$ *and* $\eta > 0$ *then*

$$\mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}} [[\![M]\!]_\eta(\boldsymbol{\theta}, \boldsymbol{s})] \xrightarrow{\text{unif.}} \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}} [[\![M]\!](\boldsymbol{\theta}, \boldsymbol{s})] \qquad \text{as } \eta \searrow 0 \text{ for } \boldsymbol{\theta} \in \boldsymbol{\Theta}$$

Proof. Let $\epsilon > 0$. By Lemma 5.24, there exists bounded U (e.g. $\mathbf{B}_\delta(\mathbf{0}) \cap \mathrm{supp}(\boldsymbol{\mathcal{D}})$ for suitably large $\delta > 0$) satisfying

$$\mathbb{E}_{\boldsymbol{z} \sim \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}} [[\boldsymbol{z} \notin U] \cdot p(\boldsymbol{z})] < \frac{\epsilon}{6}$$

for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. Besides, since $\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z})$ is continuous, we can define

$$d \coloneqq \sup_{(\boldsymbol{\theta},\boldsymbol{z})\in\boldsymbol{\Theta}\times U} |\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p(\boldsymbol{z})|$$

By uniform almost uniform convergence there exists $U' \subseteq U$ and $\eta_0 > 0$ satisfying $\mathrm{Leb}(U') < \frac{\epsilon}{6d}$ and

$$|[\![M]\!]^t_\eta(\boldsymbol{\theta},\boldsymbol{z}) - [\![M]\!]^t(\boldsymbol{\theta},\boldsymbol{z})| < \frac{\epsilon}{3}$$

for $0 < \eta < \eta_k$ and $(\boldsymbol{\theta},\boldsymbol{z}) \in \boldsymbol{\Theta} \times (U \setminus U')$. Consequently,

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[|[\![M]\!]_\eta(\boldsymbol{\theta},\boldsymbol{s}) - [\![M]\!](\boldsymbol{\theta},\boldsymbol{s})|] \\
&\leq \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}\left[[\boldsymbol{z} \notin U] \cdot |[\![M]\!]^t_\eta(\boldsymbol{\theta},\boldsymbol{z}) - [\![M]\!]^t(\boldsymbol{\theta},\boldsymbol{z})|\right] \\
&\qquad + \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}\left[[\boldsymbol{z} \in U \setminus U'] \cdot |[\![M]\!]^t_\eta(\boldsymbol{\theta},\boldsymbol{z}) - [\![M]\!]^t(\boldsymbol{\theta},\boldsymbol{z})|\right] \\
&\qquad + \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}\left[[\boldsymbol{z} \in U'] \cdot |[\![M]\!]^t_\eta(\boldsymbol{\theta},\boldsymbol{z}) - [\![M]\!]^t(\boldsymbol{\theta},\boldsymbol{z})|\right] \\
&\leq \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}\left[[\boldsymbol{z} \notin U] \cdot 2 \cdot p(\boldsymbol{z})|\right] \\
&\qquad + \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}\left[[\boldsymbol{z} \in U \setminus U'] \cdot |[\![M]\!]^t_\eta(\boldsymbol{\theta},\boldsymbol{z}) - [\![M]\!]^t(\boldsymbol{\theta},\boldsymbol{z})|\right] \\
&\qquad + \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}\left[[\boldsymbol{z} \in U'] \cdot 2 \cdot p(\boldsymbol{z})\right] \\
&\leq \frac{\epsilon}{3} + \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}\left[[\boldsymbol{z} \in U \setminus U'] \cdot \frac{\epsilon}{3}\right] + 2d \cdot \mathrm{Leb}(U') \\
&\leq \epsilon
\end{aligned}$$

□

Finally, by Remark 5.13, the polynomial bound in the premise can be established with the $\vdash_{\mathrm{int}}$ type system of Section 4.5.1:

Theorem 5.36. *If* $\theta_1 : \iota_1^{(\mathbf{f},\emptyset)}, \ldots, \theta_m : \iota_m^{(\mathbf{f},\emptyset)} \mid \Sigma \vdash_{\mathrm{unif}} M : R^{(\sigma,\mathrm{dep})}$ *and (also assuming Assumption 5.9)* $\theta_1 : \iota_1^{(\mathbf{c})}, \ldots, \theta_m : \iota_m^{(\mathbf{c})} \mid \Sigma \vdash_{\mathrm{int}} M : R^{(\mathbf{p})}$ *then*

$$\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[[\![M]\!]_\eta(\boldsymbol{\theta},\boldsymbol{s})\right] \xrightarrow{\text{unif.}} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[[\![M]\!](\boldsymbol{\theta},\boldsymbol{s})\right] \qquad \text{as } \eta \searrow 0 \text{ for } \boldsymbol{\theta} \in \boldsymbol{\Theta}$$

We demonstrate Proposition 5.34 by positing the following infinitary logical relation $\mathcal{UAU}^{(n)}_\tau$, $n \in \mathbb{N}$, where $n \in \mathbb{N}$, between sequences (index by $\eta > 0$) of morphisms $\boldsymbol{\Theta} \times \mathbb{R}^n \to [\![\tau]\!]$ in **Fr** (corresponding to the smoothings) and $\boldsymbol{\Theta} \times \mathbb{R}^n \to [\![\tau]\!]$ in **QBS** (corresponding to the measurable standard semantics).

(i) $(f_\eta, f) \in \mathcal{UAU}^{(n)}_{\iota(\sigma,\mathrm{dep})}$ if $f_\eta \xrightarrow{\text{u.a.u.}} f$ and for all bounded $U \subseteq \boldsymbol{\Theta} \times \mathbb{R}^n$, $f(U)$ is bounded.

(ii) $(f_\eta, f) \in \mathcal{UAU}^{(n)}_{\tau_1 \bullet \Sigma_3 \to \tau_2}$ if for all $n' \in \mathbb{N}$ and $(g_\eta, g) \in \mathcal{UAU}^{(n+n')}_{\tau_1}$,

$$(f_\eta \odot g_\eta, f \odot g) \in \mathcal{UAU}^{(n+n'+|\Sigma_3|)}_{\tau_2}$$

We then prove a fundamental lemma:

Lemma 5.37 (Fundamental)**.** *If* $\theta_1 : \iota_1^{(\mathbf{f},\emptyset)}, \ldots, \theta_m : \iota_m^{(\mathbf{f},\emptyset)}, x_1 : \tau_1, \ldots, x_\ell : \tau_\ell \mid \Sigma \vdash_{\mathrm{unif}} M : \tau$, $n \in \mathbb{N}$, $(\xi^{(1)}_\eta, \xi^{(1)}) \in \mathcal{UAU}^{(n)}_{\tau_1}, \ldots, (\xi^{(\ell)}_\eta, \xi^{(\ell)}) \in \mathcal{UAU}^{(n)}_{\tau_\ell}$ *then*

$$\left([\![M]\!]^t_\eta * \langle \xi^{(1)}_\eta, \ldots, \xi^{(\ell)}_\eta \rangle, [\![M]\!]^t * \langle \xi^{(1)}, \ldots, \xi^{(\ell)} \rangle\right) \in \mathcal{UAU}^{(n+|\Sigma|)}_\tau$$

where $([\![M]\!]^t * \langle \xi^{(1)}, \ldots, \xi^{(\ell)} \rangle)(\boldsymbol{\theta}, \boldsymbol{s} +\!\!+ \boldsymbol{s}') \coloneqq [\![M]\!]^t(\boldsymbol{\theta}, (\xi^{(1)}(\boldsymbol{\theta},\boldsymbol{s}), \ldots, \xi^{(\ell)}(\boldsymbol{\theta},\boldsymbol{s})), \boldsymbol{s}')$ *and similarly for* $[\![M]\!]^t_\eta$.

The case for primitive operations exploits the following, which is proven in Appendix B.2:

Lemma 5.38. *Let $f : U_1 \times \cdots \times U_\ell \to \mathbb{R}$ (for open and connected $U_1, \ldots, U_\ell \subseteq \mathbb{R}$) and $g_\eta^{(i)}, g^{(i)} : \mathbf{\Theta} \times \mathbb{R}^n \to U_i$ for $1 \leq i \leq \ell$ and $\eta > 0$. Suppose the following holds*

(i) *f is continuously differentiable*
(ii) *$g^{(i)}(U)$ is bounded for $1 \leq i \leq n$ and all bounded $U \subseteq \mathbf{\Theta} \times \mathbb{R}^n$*
(iii) *$g_\eta^{(i)} \xrightarrow{\text{u.a.u.}} g^{(i)}$ for $1 \leq i \leq n$*

Then $f \circ \langle g_\eta, h_\eta \rangle \xrightarrow{\text{u.a.u.}} f \circ \langle g, h \rangle : \mathbf{\Theta} \times \mathbb{R}^n \to \mathbb{R}$.

Proof of Lemma 5.37. The claim is proven by induction on the typing judgements. We focus on the most interesting cases:

– By Lemma 5.30, the guard of conditionals cannot depend on the parameters and is different from 0 a.e. Therefore, u.a.u. convergence for conditionals follows from Lemmas 5.21(i) and 5.38 and Remark 5.32. The boundedness condition is obvious.

– For primitive operations uniform almost everywhere convergence follows from the inductive hypothesis and Lemma 5.38. The boundedness condition follows from the assumption that primitives are analytic (hence continuous). □

Discussion

The type system is incomplete, in the sense that there are terms-in-context that satisfy the uniform convergence property of the objective function but which are not typable.

Example 5.39 (Incompleteness)**.** (i) The following term-in-context is not typable because (2 being constant) $\not\vdash_{\text{unif}} 2 : R^{(\mathbf{t},\emptyset)}$:

$$| \, [\mathcal{N}] \not\vdash_{\text{unif}} \mathbf{if}\, 2 \cdot (\mathbf{sample}_{\mathcal{N}} \rhd \mathrm{id}) < 0 \,\mathbf{then}\, 0 \,\mathbf{else}\, 1 : R^{(\sigma,\text{dep})}$$

However, the impact of this can be reduced by admitting more primitives: if $f(x) := 2 \cdot x$ is in Op then

$$| \, [\mathcal{N}] \not\vdash_{\text{unif}} \mathbf{if}\, \underline{f}(\mathbf{sample}_{\mathcal{N}} \rhd \mathrm{id}) < 0 \,\mathbf{then}\, 0 \,\mathbf{else}\, 1 : R^{(\mathbf{f},\emptyset)}$$

(ii) The following term-in-context denotes the "identity":

$$| \, [] \vdash_{\text{unif}} (\lambda x.(2 \cdot x) - x) : R^{(\mathbf{t},\{\alpha_1\})} \to R^{(\mathbf{f},\{\alpha_1\})}$$

but it does *not* have type $R^{(\mathbf{t},\{\alpha_1\})} \to R^{(\mathbf{t},\{\alpha_1\})}$. Then, using the same reasoning as Example 5.29, the term

$$G \equiv (\lambda x.(2 \cdot x) - x)\,(\mathbf{sample}_{\mathcal{D}} \rhd \mathrm{id})$$

has type $R^{(\mathbf{f},\{\alpha_1\})}$, but *not* $R^{(\mathbf{t},\{\alpha_1\})}$, and so $\mathbf{if}\, G < 0 \,\mathbf{then}\, 0 \,\mathbf{else}\, 1$ is not typable, even though G can safely be used in guards.

5.4 Conclusion and Related Work

We have endowed our language with a smoothed semantics approximating potentially discontinuous programs, which is parameterised by an accuracy coefficient. We have proposed type systems to guarantee pleasing properties in the context of stochastic optimisation: For a fixed accuracy coefficient, stochastic gradient descent converges to stationary points even with the reparameterisation gradient (which is *unbiased*). Besides, the smoothed objective function converges uniformly to the true objective as the accuracy is improved.

Our type systems can be used to *independently* check these two properties ($\vdash_{\text{int}}$ for the correctness of SGD for each smoothing, $\vdash_{\text{unif}}$ for the convergence of approximations) to obtain partial theoretical guarantees even if one of the systems suffers from incompleteness. We also stress that SGD and the smoothed unbiased gradient estimator can even be applied to programs which are *not* typable (whilst sacrificing or manually proving the guarantees).

Related Work

[Lee et al., 2018] is the starting point for our work. They correct the (biased) reparameterisation gradient estimator for non-differentiable models by additional non-trivial *boundary* terms. They present an efficient method for *affine* guards only. Besides, they are not concerned with the *convergence* of gradient-based optimisation procedures; nor do they discuss how assumptions they make may be manifested in a programming language.

In practice, non-differentiable functions are often approximated smoothly without placing emphasis on theoretical guarantees. In the context of the reparameterisation gradient, [Maddison et al., 2017, Jang et al., 2017] relax discrete random variables in a continuous way, effectively dealing with a specific class of discontinuous models. [Tucker et al., 2017] combine the score estimator with *control variates* (a common technique in statistics to reduce variances) based on such continuous approximations in a clever way. Some foundations of smoothing approaches are studied in [Bertsekas, 1975, Zang, 1981] in a non-stochastic setting.

Motivated by parameter synthesis, [Chaudhuri and Solar-Lezama, 2010, Chaudhuri and Solar-Lezama, 2011] seem to be the first to discuss systematic smoothing approaches in the context of programming languages. They do not study *stochastic* optimisation and thus consider a purely deterministic, imperative language. They use a very different approach to smoothing, Gaussian smoothing, which for an input yields the expectation of the (standard) denotation applied to that input perturbed by a Gaussian noise. As such, it does not exploit the fact that the function is represented in a programming language. Furthermore, they only present *approximate* methods to compute the smoothing.

In as yet unpublished work, [de Amorim and Lam, 2022] propose an alternative approach to circumvent discontinuities drawing on ideas from distribution theory. Their language does not allow nesting conditionals and is purely deterministic. They do include non-standard constructs which amount to *approximately* defining integrals on *compact* sets.

As an application to their work on semantics of functional programming languages with conditionals, [Huot et al., 2023] study the correctness of gradient descent for *deterministic* programs and do not deal with *stochastic* optimisation.

[Gorinova et al., 2022] investigate a variant of the imperative probabilistic programming language Stan, and they present an information flow type system to detect conditional independence relations. Their main application is the efficient elimination of discrete parameters by explicitly marginalising them out. To capture conditional independence relations, they annotate types with one of three levels and ensure that only certain dependence relations hold between these levels. Our $\vdash_{\text{unif}}$-type system in Section 5.3.2 has a completely different objective (ensuring guards are almost everywhere not 0). Besides, our annotations $\iota^{(\sigma,\text{dep})}$ are in a sense more fine-grained because we need to capture dependencies on specific samples rather than just their (one of three) levels.

Two closely related works have been developed concurrently to our approach: [Lee et al., 2023, Lew et al., 2023]. Both study gradient estimators for probabilistic programming. However, a key difference is that their aim is to identify (for each sample) which of the estimators (reparameterisation or score) can be applied safely. Our work on the other hand is complementary: it studies approaches to employ smoothing and the reparameterisation estimator when it would otherwise *not* be applicable (and hence their systems refuse using the reparameterisation estimator at least for some of the variables). Furthermore, they "only" study the bias of gradient estimators and not the correctness of the overall (SGD) optimisation procedure.

[Lee et al., 2023] study an imperative language inspired by Pyro, and they use abstract interpretation [Cousot and Cousot, 1977] to spot smoothness properties.

[Lew et al., 2023] present an intriguing program transformation, which translates (higher-order) functional programs to new probabilistic programs, the expectation of which is the derivative of the expectation of the original program. They use a lightweight type system with annotations (resembling our approach in Section 4.6) to track values which may be used non-smoothly to identify when the reparameterisation estimator may be correctly employed. (Otherwise, in contrast to our work, they only permit the Score estimator. Furthermore, they do not automatically check the domination premise of Lemma 2.17 required to fully justify unbiasedness.)

Chapter 6

Diagonalisation Stochastic Gradient Descent

A natural question to ask is: *how can we choose an accuracy coefficient for a "decent" approximation of the original objective function?* For our running Example 2.16 we can observe that this really matters: stationary points for low accuracy (i.e. high η) may not yield significantly better results than the biased (standard) reparameterisation gradient estimator (see Fig. 6.1).

In the preceding chapter we have conducted a rudimentary analysis of the relationship between the η-smoothed and the original problem (4.12): we have shown that under mild syntactic restrictions $\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[[\![M]\!]_\eta(\boldsymbol{\theta},\boldsymbol{s})]$ converges uniformly to $\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[[\![M]\!](\boldsymbol{\theta},\boldsymbol{s})]$ as $\eta \searrow 0$. As a consequence, for any given error tolerance $\epsilon > 0$, there *exists* an accuracy coefficient η such that the η-smoothed and true objective functions only differ by at most ϵ.

However, the practical relevance of this result is limited: it is generally not clear how to extract the accuracy coefficient from the convergence proof. Besides, we may wish to improve the accuracy of the approximation, e.g. once we have found a stationary point to an approximation that proves to be a relatively poor approximation to the true objective function. Moreover, unfortunately, there is no bound (as $\eta \searrow 0$) to the derivative of σ_η at 0 (see Fig. 5.1b). Therefore, the variance of the smoothed estimator also increases as the accuracy is enhanced, which may result in slow or unstable convergence of SGD.

We offer a solution with theoretical guarantees for these issues in the present chapter. In particular, we extend our work in the following ways:

- We introduce the DSGD algorithm, and show that gradients for the *original* (unsmoothed) problem vanish asymptotically. Crucially, the accuracy coefficient does not need to be fixed in advance; rather it is progressively enhanced during the optimisation (which has important advantages).
- We identify syntactic criteria to establish pre-conditions of the algorithm. In particular, we bound the variance in a purely syntactic manner based on the depth of nesting of conditionals into guards.

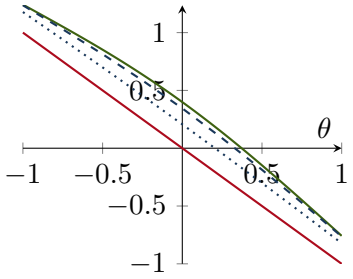


Figure 6.1: Solid red: biased estimator $\mathbb{E}_{z\sim\mathcal{N}(0,1)}[\nabla_\theta f(\theta, z)]$, solid green: true gradient $\nabla_\theta \mathbb{E}_{z\sim\mathcal{N}(0,1)}[f(\theta, z)]$, black: gradient of smoothed objective (dotted: $\eta = 1$, dashed: $\eta = 1/3$) for Example 2.16.

6.1 Diagonalisation SGD

We propose a novel variant of SGD in which we gradually enhance the accuracy coefficient *during* optimisation (rather than *fixing* it in advance). To streamline the presentation we focus on the Simple Function Calculus Optimisation Problem 4.14 in this chapter.

Thus, we fix a compact parameter space $\boldsymbol{\Theta} \subseteq \mathbb{R}^m$, $\mathcal{D}_1, \ldots, \mathcal{D}_n \in \mathrm{Dist}$ as well as $\varphi^{(1)}_{(-)} \in \mathrm{Diff}_{\boldsymbol{\Theta}}(\mathrm{supp}(\mathcal{D}_1)), \ldots, \varphi^{(n)}_{(-)} \in \mathrm{Diff}_{\boldsymbol{\Theta}}(\mathrm{supp}(\mathcal{D}_n))$, and we recall the abbreviations

$$\boldsymbol{\mathcal{D}}(\boldsymbol{s}) \coloneqq \prod_{i=1}^{n} \mathcal{D}_i(s_i) \qquad\qquad \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \coloneqq \left(\varphi^{(1)}_{\boldsymbol{\theta}}(s_1), \ldots, \varphi^{(n)}_{\boldsymbol{\theta}}(s_n)\right)$$

For a term $F \in \Lambda^{\mathrm{SFC}}$ and a sequence of accuracy coefficients $\eta_k \searrow 0$, we modify the standard stochastic gradient descent iteration (SGD') to

$$\boldsymbol{\theta}_{k+1} \coloneqq \boldsymbol{\theta}_k - \gamma_k \cdot \nabla_{\boldsymbol{\theta}} \, [\![F]\!]_{\eta_k}(\boldsymbol{\varphi}_{\boldsymbol{\theta}_k}(\boldsymbol{s}_k)) \qquad\qquad \boldsymbol{s}_k \sim \boldsymbol{\mathcal{D}} \tag{DSGD'}$$

Intuitively, this scheme facilitates getting close to the optimum whilst the variance is low (but the approximation may be coarse) and make small adjustments once the accuracy has been enhanced and approximation errors become visible.

To formalise a correctness result, we generalise the setting: suppose for each $k \in \mathbb{N}$, $f_k : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$ is differentiable. We define a ***Diagonalisation Stochastic Gradient Descent (DSGD)*** sequence:

$$\boldsymbol{\theta}_{k+1} \coloneqq \boldsymbol{\theta}_k - \gamma_k \cdot \nabla_{\boldsymbol{\theta}} \, f_k(\boldsymbol{\theta}_k, \boldsymbol{s}_k) \qquad\qquad \boldsymbol{s}_k \sim \boldsymbol{\mathcal{D}} \tag{DSGD}$$

where γ_k is the step size. The qualifier "diagonal" highlights that, in contrast to (SGD'), we are not using the gradient of the same function, but rather we are using the gradient of f_k in step k.

Due to the aforementioned fact that also the variance increases as the accuracy is enhanced, the scheme of accuracy coefficients $(\eta_k)_{k\in\mathbb{N}}$ needs to be adjusted carefully to tame the growth of the variances[1] V_k of the gradient of f_k, as stipulated by the following equation:

$$\sum_{k\in\mathbb{N}} \gamma_k = \infty \qquad\qquad \sum_{k\in\mathbb{N}} \gamma_k^2 \cdot V_k < \infty \tag{6.1}$$

[1] It will not be necessary to compute variances exactly. Rather, it suffices to bound their growth up to a constant (see Section 6.3).

Note that in the regime $f_k = f$ and $V_k = V$ of standard SGD, this condition subsumes the standard Robbins-Monro condition [Robbins and Monro, 1951] and $\gamma_k \in \Theta(1/k)$ can be chosen.

The following generalises the correctness result (Proposition 2.11) for standard stochastic gradient descent:

Proposition 6.1 (Correctness)**.** *Suppose $\boldsymbol{\Theta} \subseteq \mathbb{R}^m$ is convex, $(\gamma_k)_{k\in\mathbb{N}}$ and $(V_k)_{k\in\mathbb{N}}$ satisfy Eq. (6.1),*

$$g_k(\boldsymbol{\theta}) \coloneqq \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[f_k(\boldsymbol{\theta}, \boldsymbol{s})] \qquad\qquad g(\boldsymbol{\theta}) \coloneqq \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[f(\boldsymbol{\theta}, \boldsymbol{s})]$$

are well-defined and differentiable, and

(DSGD1) $\nabla_{\boldsymbol{\theta}}\, g_k(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[\nabla_{\boldsymbol{\theta}}\, f_k(\boldsymbol{\theta}, \boldsymbol{s})]$ *for all* $k \in \mathbb{N}$ *and* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$
(DSGD2) g *is bounded, Lipschitz continuous and Lipschitz smooth on* $\boldsymbol{\Theta}$
(DSGD3) $\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[\|\nabla_{\boldsymbol{\theta}}\, f_k(\boldsymbol{\theta}, \boldsymbol{s})\|^2] < V_k$ *for all* $k \in \mathbb{N}$ *and* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$
(DSGD4) ∇g_k *converges uniformly to* ∇g *on* $\boldsymbol{\Theta}$

Then almost surely[2] $\liminf_{i\to\infty} \|\nabla g(\boldsymbol{\theta}_i)\| = 0$ *or* $\boldsymbol{\theta}_i \notin \boldsymbol{\Theta}$ *for some* $i \in \mathbb{N}$.

Noteworthy differences to Proposition 2.11 include the additional premise (DSGD4) and the fact that in (DSGD3) we do not assume a bound *uniform* in $k \in \mathbb{N}$.

Proof. Let L be the constant for Lipschitz smoothness in (DSGD2). By Taylor's theorem and the convexity of $\boldsymbol{\Theta}$, for any $\boldsymbol{\theta}_k \in \boldsymbol{\Theta}$,

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{s}_k\sim\boldsymbol{\mathcal{D}}}[g(\boldsymbol{\theta}_{k+1})] \\
&\leq \mathbb{E}_{\boldsymbol{s}_k\sim\boldsymbol{\mathcal{D}}}\left[g(\boldsymbol{\theta}_k) - \gamma_k \cdot \langle \nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}_k), \nabla_{\boldsymbol{\theta}}\, f_k(\boldsymbol{\theta}_k, \boldsymbol{s}_k)\rangle + \frac{\gamma_k^2}{2} \cdot L \cdot \|\nabla_{\boldsymbol{\theta}}\, f_k(\boldsymbol{\theta}_k, \boldsymbol{s}_k)\|^2\right] \\
&\leq g(\boldsymbol{\theta}_k) - \gamma_k \cdot \langle \nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}_k), \nabla_{\boldsymbol{\theta}}\, g_k(\boldsymbol{\theta}_k)\rangle + \frac{\gamma_k^2}{2} \cdot L \cdot V_k
\end{aligned}$$

using (DSGD1) and (DSGD3) in the second step. Hence,

$$\begin{aligned}
&\gamma_k \cdot \mathbb{E}_{\boldsymbol{s}_0,\ldots,\boldsymbol{s}_{k-1}\sim\boldsymbol{\mathcal{D}}}\left[\langle \nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}_k), \nabla_{\boldsymbol{\theta}}\, g_k(\boldsymbol{\theta}_k)\rangle\right] \\
&\leq \mathbb{E}_{\boldsymbol{s}_0,\ldots,\boldsymbol{s}_{k-1}\sim\boldsymbol{\mathcal{D}}}\left[g(\boldsymbol{\theta}_k)\right] - \mathbb{E}_{\boldsymbol{s}_0,\ldots,\boldsymbol{s}_k\sim\boldsymbol{\mathcal{D}}}[g(\boldsymbol{\theta}_{k+1})] + \frac{\gamma_k^2}{2} \cdot L \cdot V_k
\end{aligned}$$

and thus,

$$\begin{aligned}
&\left(\sum_{i=0}^{k} \gamma_i\right) \cdot \min_{i=0}^{k} \mathbb{E}_{\boldsymbol{s}_0,\ldots,\boldsymbol{s}_{i-1}\sim\boldsymbol{\mathcal{D}}}[\langle \nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}_i), \nabla_{\boldsymbol{\theta}}\, g_i(\boldsymbol{\theta}_i)\rangle] \\
&\leq \sum_{i=0}^{k} \gamma_i \cdot \mathbb{E}_{\boldsymbol{s}_0,\ldots,\boldsymbol{s}_{i-1}\sim\boldsymbol{\mathcal{D}}}[\langle \nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}_i), \nabla_{\boldsymbol{\theta}}\, g_i(\boldsymbol{\theta}_i)\rangle] \\
&\leq g(\boldsymbol{\theta}_0) - \mathbb{E}_{\boldsymbol{s}_0,\ldots,\boldsymbol{s}_k\sim\boldsymbol{\mathcal{D}}}[g(\boldsymbol{\theta}_{k+1})] + L \cdot \sum_{i=0}^{k} \frac{\gamma_i^2}{2} \cdot V_i
\end{aligned}$$

[2] w.r.t. the random choices $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots$ of DSGD

By boundedness (DSGD2), $g(\boldsymbol{\theta}_0) - \mathbb{E}[g(\boldsymbol{\theta}_{k+1})] \leq c < \infty$ (for all $k \in \mathbb{N}$ for some $c > 0$) and therefore due to Eq. (6.1) it follows:

$$\inf_{i\in\mathbb{N}} \mathbb{E}[\langle \nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}_i), \nabla_{\boldsymbol{\theta}}\, g_i(\boldsymbol{\theta}_i)\rangle] \leq \left(\sum_{i\in\mathbb{N}} \gamma_i\right)^{-1} \cdot \left(c + L \cdot \sum_{i\in\mathbb{N}} \frac{\gamma_i^2}{2} \cdot V_i\right) = 0$$

By the same reasoning for all $i_0 \in \mathbb{N}$, $\inf_{i\geq i_0} \mathbb{E}[\langle \nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}_i), \nabla_{\boldsymbol{\theta}}\, g_i(\boldsymbol{\theta}_i)\rangle] = 0$ and therefore also $\liminf_{i\in\mathbb{N}} \mathbb{E}[\langle \nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}_i), \nabla_{\boldsymbol{\theta}}\, g_i(\boldsymbol{\theta}_i)\rangle] = 0$.

Next, observe that

$$\begin{aligned}\|\nabla\, g(\boldsymbol{\theta}_i)\|^2 &= \langle \nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}_i), \nabla_{\boldsymbol{\theta}}\, g_i(\boldsymbol{\theta}_i)\rangle + \langle \nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}_i), \nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}_i) - \nabla_{\boldsymbol{\theta}}\, g_i(\boldsymbol{\theta}_i)\rangle \\ &\leq \langle \nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}_i), \nabla_{\boldsymbol{\theta}}\, g_i(\boldsymbol{\theta}_i)\rangle + \|\nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}_i)\| \cdot \|\nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}_i) - \nabla_{\boldsymbol{\theta}}\, g_i(\boldsymbol{\theta}_i)\|\end{aligned}$$

and the right summand uniformly converges to 0 because of Lipschitz continuity (DSGD2) and the fact that $\|\nabla_{\boldsymbol{\theta}}\, g(\boldsymbol{\theta}_i) - \nabla_{\boldsymbol{\theta}}\, g_i(\boldsymbol{\theta}_i)\|$ does so by premise (DSGD4). Consequently,

$$\liminf_{i\in\mathbb{N}} \mathbb{E}\left[\|\nabla\, g(\boldsymbol{\theta}_i)\|^2\right] = 0$$

Finally, this can only possibly hold if *almost surely* $\liminf_{i\in\mathbb{N}} \|\nabla\, g(\boldsymbol{\theta}_i)\|^2 = 0$. □

6.2 Establishing Pre-Conditions

Next, we wish to show that Proposition 6.1 is applicable to $f_k \coloneqq [\![F]\!]_{\eta_k} \circ \boldsymbol{\varphi}_{(-)}$ for a suitable scheme of accuracy coefficients $(\eta_k)_{k\in\mathbb{N}}$. Having already demonstrated unbiasedness (DSGD1) in Theorem 5.15, we proceed by addressing the remaining proof obligations. We recall our assumption on the primitives for the simple function calculus:

Assumption 6.2. Op is a set of smooth functions bounded by polynomials, partial derivatives of which up to order 2 are bounded by polynomials.

Similarly to Lemma 5.24 we can prove (see Appendix C.1 for details):

Lemma 6.3. *If $f : U \to \mathbb{R}$ is a Schwartz function, $\boldsymbol{\varphi}_{(-)} : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}^n$ is a polynomial and a strong diffeomorphism, and $p : \mathbb{R}^n \to \mathbb{R}$ is a polynomial then*

$$\begin{aligned}\int_U \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left|\frac{\partial f_{(-)}}{\partial \theta_i}(\boldsymbol{\theta}, \boldsymbol{z}) \cdot p(\boldsymbol{z})\right| \mathrm{d}\boldsymbol{z} < \infty \\ \int_U \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left|\frac{\partial f_{(-)}}{\partial z_i}(\boldsymbol{\theta}, \boldsymbol{z}) \cdot p(\boldsymbol{z})\right| \mathrm{d}\boldsymbol{z} < \infty \\ \int_U \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left|\frac{\partial^2 f_{(-)}}{\partial \theta_i \partial \theta_j}(\boldsymbol{\theta}, \boldsymbol{z}) \cdot p(\boldsymbol{z})\right| \mathrm{d}\boldsymbol{z} < \infty\end{aligned}$$

where $f_{\boldsymbol{\theta}}(\boldsymbol{z}) \coloneqq f(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) \cdot |\det \mathbf{J}\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})|$.

Thus, to show Lipschitz smoothness (DSGD2) of $[\![F]\!]$ (for $F \in \Lambda^{\mathrm{SFC}}$) we may exchange differentiation and integration (justified by Lemmas 2.17 and 2.24 since $[\![F]\!]$ does not depend on parameters):

$$\left|\frac{\partial^2}{\partial \theta_i \partial \theta_j} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[[\![F]\!](\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))]\right| = \left|\frac{\partial^2}{\partial \theta_i \partial \theta_j} \mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}[[\![F]\!](\boldsymbol{z})]\right|$$

$$\leq \int_{\mathrm{supp}(\boldsymbol{\mathcal{D}})} \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left| \left(\frac{\partial^2}{\partial\theta_i \partial\theta_j} \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{z}) \right) \cdot [\![F]\!](\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z} < \infty$$

for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and we conclude:

Corollary 6.4 (Lipschitz Smoothness)**.** *If $F \in \Lambda^{\mathrm{SFC}}$ then the function*

$$\boldsymbol{\theta} \mapsto \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[[\![F]\!](\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))]$$

is Lipschitz smooth.

In the same manner we can prove the other (simpler) obligations of (DSGD2). Furthermore, exploiting Lemma 6.3, we can prove similarly to Corollary 5.23 that not only the smoothed objective function but also the gradients converge uniformly (DSGD4); see Appendix C.1 for details:

Proposition 6.5. *If $F \in \Lambda^{\mathrm{SFC}}_{\mathrm{s}}$ then*

$$\nabla_{\boldsymbol{\theta}}\, \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[[\![F]\!]_\eta(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))] \xrightarrow{\text{unif.}} \nabla_{\boldsymbol{\theta}}\, \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[[\![F]\!](\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))] \qquad \text{as } \eta \searrow 0 \text{ for } \boldsymbol{\theta} \in \boldsymbol{\Theta}$$

6.2.1 Bounding the Variance

Next, we analyse the variance for (DSGD3). Recall that nesting if-statements in guards (e.g. F_2 in Example 4.13) results in nestings[3] of σ_η in the smoothed interpretation, which in view of the chain rule[4] may cause high variance as $\sigma'_\eta(0) = \frac{1}{4\eta}$ for the logistic sigmoid. Therefore, to give good bounds we consider subsets $\Lambda^{\mathrm{SFC}}_\ell$ of Λ^{SFC} in which ℓ is the maximal nesting depth of the guards of if-statements.

Formally, we define $\Lambda^{\mathrm{SFC}}_\ell$ inductively:

$$\frac{}{z_j \in \Lambda^{\mathrm{SFC}}_0} \qquad \frac{F_1 \in \Lambda^{\mathrm{SFC}}_\ell \quad \cdots \quad F_k \in \Lambda^{\mathrm{SFC}}_\ell}{f(F_1,\ldots,F_k) \in \Lambda^{\mathrm{SFC}}_\ell}\ f : \mathbb{R}^k \to \mathbb{R} \in \mathrm{Op}$$

$$\frac{F \in \Lambda^{\mathrm{SFC}}_\ell}{F \in \Lambda^{\mathrm{SFC}}_{\ell+1}} \qquad \frac{F \in \Lambda^{\mathrm{SFC}}_\ell \quad G \in \Lambda^{\mathrm{SFC}}_{\ell+1} \quad H \in \Lambda^{\mathrm{SFC}}_{\ell+1}}{(\mathbf{if}\ F < 0\ \mathbf{then}\ G\ \mathbf{else}\ H) \in \Lambda^{\mathrm{SFC}}_{\ell+1}}$$

Note that $\Lambda^{\mathrm{SFC}} = \bigcup_{\ell\in\mathbb{N}} \Lambda^{\mathrm{SFC}}_\ell$. For the expressions in Example 4.13, $F_1, F'_1 \in \Lambda^{\mathrm{SFC}}_1$ and $F_2 \in \Lambda^{\mathrm{SFC}}_2$.

Now, exploiting the chain rule, Assumption 6.2 and Lemma 2.25(i), it is straightforward to show inductively that for $F \in \Lambda^{\mathrm{SFC}}_\ell$,

Lemma 6.6. *If $F \in \Lambda^{\mathrm{SFC}}_\ell$ there exists a polynomial $p : \mathbb{R}^n \to \mathbb{R}$ such that for all $\eta > 0$ and $\boldsymbol{z} \in \mathbb{R}^n$,*

$$\left| \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z}) \right| \leq |\sigma'_\eta|^\ell \cdot p(\boldsymbol{z})$$

(By $|\sigma'_\eta|$ we mean $\sup_{y\in\mathbb{R}} |\sigma'_\eta(y)|$.) In particular, the smaller the accuracy coefficient $\eta > 0$ is, the higher the bound on the variance is. Consequently:

[3] as in $\sigma_\eta(\sigma_\eta(\cdots)\cdots)$

[4] which yields products of σ'_η

Proposition 6.7. *If $F \in \Lambda_\ell^{\mathrm{SFC}}$ then there exists $c > 0$ such that for all $\eta > 0$ and $\boldsymbol{\theta} \in \boldsymbol{\Theta}$,*

$$\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[\|\nabla_{\boldsymbol{\theta}}[\![F]\!]_\eta(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))\|^2\right] \leq c \cdot |\sigma_\eta'|^{2\ell}$$

We can give a sharper bound using slightly stronger assumptions on σ_η and the support of the permissible distributions in Dist, which will allows us to enhance the accuracy more rapidly (in view of Eq. (6.1)):

Assumption 6.8. (i) For all $\mathcal{D} \in \mathrm{Dist}$, $\mathrm{supp}(\mathcal{D}) = \mathbb{R}$ or $\mathrm{supp}(\mathcal{D}) = \mathbb{R}_{\geq 0}$.
(ii) $\sigma_\eta(x) \coloneqq \sigma(\frac{x}{\eta})$, where σ is the (logistic) sigmoid function $\sigma(x) \coloneqq \frac{1}{1+\exp(-x)}$.

Proposition 6.9. *If $F \in \Lambda_\ell^{\mathrm{SFC}}$ then there exists $c > 0$ such that for all $\eta > 0$ and $\boldsymbol{\theta} \in \boldsymbol{\Theta}$,*

$$\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[\|\nabla_{\boldsymbol{\theta}}[\![F]\!]_\eta(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))\|^2] \leq c \cdot \eta^{-\ell}$$

Note that for the logistic sigmoid, $|\sigma_\eta'(x)| \leq \eta^{-1}$.
To get an intuitive idea of the proof we consider our running example

$$F_1' \equiv (\mathbf{if}\, z < 0\, \mathbf{then}\, 0\, \mathbf{else}\, 1) \in \Lambda_1^{\mathrm{SFC}}$$

from Example 4.13 and $\varphi_\theta(s) = s + \theta$.

$$\begin{aligned}
&\mathbb{E}_{s\sim\mathcal{N}}\left[\left|\frac{\partial([\![F_1']\!]_\eta \circ \varphi_{(-)})}{\partial\theta}(\theta, s)\right|^2\right] \\
&= \mathbb{E}_{s\sim\mathcal{N}}\left[\left(\sigma_\eta'(\varphi_\theta(s))\right)^2\right] \\
&\leq |\sigma_\eta'| \cdot \int \mathcal{N}(s) \cdot \frac{\partial(\sigma_\eta \circ \varphi_{(-)})}{\partial s}(\theta, s)\,\mathrm{d}s \\
&= |\sigma_\eta'| \cdot \left(\underbrace{[\mathcal{N}(s) \cdot \sigma_\eta(\varphi_\theta(s))]_{-\infty}^{\infty}}_{=0} - \int \mathcal{N}'(s) \cdot \sigma_\eta(\varphi_\theta(s))\,\mathrm{d}s\right) \\
&\leq |\sigma_\eta'| \cdot \int |\underbrace{\mathcal{N}'(s)}_{\text{Schwartz}}|\,\mathrm{d}s
\end{aligned}$$

where we used integration by parts in the penultimate step.

Proof of Proposition 6.9. Note that

$$\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[\left|\frac{\partial([\![F]\!]_\eta \circ \boldsymbol{\varphi}_{(-)})}{\partial\theta_i}(\boldsymbol{s})\right|^2\right] = \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[\left|\sum_{j=1}^{n} \frac{\partial[\![F]\!]_\eta}{\partial z_j}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \cdot \frac{\partial\varphi_{(-)}^{(j)}}{\partial\theta_i}(\boldsymbol{\theta}, \boldsymbol{s})\right|^2\right]$$

Therefore, by the Cauchy-Schwarz inequality it suffices to bound

$$\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}\left[\left|\frac{\partial[\![F]\!]_\eta}{\partial z_j}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{z})) \cdot \frac{\partial\varphi_{(-)}^{(j)}}{\partial\theta_i}(\boldsymbol{\theta}, \boldsymbol{s})\right|^2\right]$$

for all $1 \leq j \leq n$. By Lemma 2.26, for each $1 \leq i \leq m$ and $1 \leq j \leq n$ there exists a (non-negative) polynomial bound $p : \mathbb{R}^n \to \mathbb{R}$ satisfying

$$\left(\frac{\partial \varphi^{(j)}_{(-)}}{\partial \theta_i}(\boldsymbol{\theta}, \boldsymbol{s}) \right)^2 \leq p(\boldsymbol{s})$$

and for $f(\boldsymbol{s}) \coloneqq \boldsymbol{\mathcal{D}}(\boldsymbol{s}) \cdot p(\boldsymbol{s})$, which is a Schwartz function by Lemma 2.20(iii),

$$\begin{aligned} \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}} \left[\left| \frac{\partial [\![F]\!]_\eta}{\partial z_j}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{z})) \cdot \frac{\partial \varphi^{(j)}_{(-)}}{\partial \theta_i}(\boldsymbol{\theta}, \boldsymbol{s}) \right|^2 \right] &\leq \int f(\boldsymbol{s}) \cdot \left| \frac{\partial [\![F]\!]_\eta}{\partial z_j}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{z})) \right|^2 \mathrm{d}\boldsymbol{s} \\ &= \int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot \left| \frac{\partial [\![F]\!]_\eta}{\partial z_j}(\boldsymbol{z}) \right|^2 \mathrm{d}\boldsymbol{z} \end{aligned}$$

where

$$f_{\boldsymbol{\theta}} \coloneqq f(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) \cdot |\det \mathbf{J} \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})|$$

The claim follows with the following lemma proven in Appendix C.1.1. □

Lemma 6.10. *Let $F \in \Lambda^{\mathrm{SFC}}_\ell$, $f : U \to \mathbb{R}$ be a non-negative Schwartz function, where $U = U_1 \times \cdots \times U_n$ and $U_1, \ldots, U_n \in \{\mathbb{R}, \mathbb{R}_{\geq 0}\}$, and p be a non-negative polynomial. For all $1 \leq i \leq n$ there exists $c > 0$ such that for all $0 < \eta \leq 1$,*

$$\int_U f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p(\boldsymbol{z}) \cdot \left| \frac{\partial^2 [\![F]\!]_\eta}{\partial z_i^2}(\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z} < c \cdot \eta^{-\ell}$$

$$\int_U f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p(\boldsymbol{z}) \cdot \left| \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z}) \right|^2 \mathrm{d}\boldsymbol{z} < c \cdot \eta^{-\ell}$$

where $f_{\boldsymbol{\theta}} \coloneqq f(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) \cdot |\det \mathbf{J} \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})|$.

6.3 Concluding Correctness

Having bounded the variance, we present a scheme of accuracy coefficients compatible with the scheme of step sizes $\gamma_k = 1/k$, which is the classic choice for SGD. Note that for any $\epsilon > 0$,

$$\sum_{k \in \mathbb{N}} \frac{1}{k} = \infty \qquad \sum_{k \in \mathbb{N}} \frac{1}{k^2} \cdot \left(k^{\frac{1}{\ell} - \epsilon} \right)^\ell < \infty \tag{6.2}$$

Therefore, with Propositions 6.1, 6.7 and 6.9 we conclude the correctness of DSGD for smoothings:

Theorem 6.11 (Correctness of DSGD)**.** *Let $F \in \Lambda^{\mathrm{SFC}}_\ell \cap \Lambda^{\mathrm{SFC}}_{\mathrm{s}}$ and $\epsilon > 0$.*
(DSGD') is correct for $\gamma_k \in \Theta(1/k)$ and $\eta_k \in \Theta(k^{-\frac{1}{\ell} + \epsilon})$:
almost surely $\liminf_{i \to \infty} \|\nabla \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}}[[\![F]\!](\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))]\| = 0$ or $\boldsymbol{\theta}_i \notin \boldsymbol{\Theta}$ for some $i \in \mathbb{N}$.

For instance for $F \in \Lambda_1^{\mathrm{SFC}}$ we can choose $\eta_k \in \Theta(1/\sqrt{k})$. Crucially, for this choice only the syntactic structure (i.e. nesting depth) of terms is essential. In particular, there is no need to calculate bounds on the Hessian, the constant in Proposition 6.9, etc.

Remark 6.12 (Choice of ϵ). Whilst asymptotically the selection of $\epsilon > 0$ does not matter in theory, in practice the choice has an effect on performance in a (necessarily finite) optimisation trajectory. Empirically, we found the values reported in Section 7.2 to work well.

The choice of ϵ needs to make a trade-off between two phenomena: smaller ϵ means faster convergence of approximation quality, whilst larger ϵ causes the (finite) variance term in Eq. (6.2) to be smaller, which in view of the proof of Proposition 6.1 is beneficial for small norms of gradients.

Remark 6.13 (Theoretical Comparison of Variances). Proposition 6.9 provides an important insight for comparing the variances of SGD for a fixed accuracy coefficient (referred to as Fixed henceforth) and DSGD: the variance is governed by the accuracy coefficient (along with the structure of the expression).

Thus, the variance of Fixed stays the same throughout a run of SGD, whereas the variance of DSGD increases over time (trading this carefully for the improvement in approximation accuracy). In particular, Proposition 6.9 yields that for time steps k, for which $\eta < \eta_k$, the bound on DSGD's variance is lower.

Rather than comparing the variance for a specific step, it is more interesting to compare the *average* variances for a trajectory of length N. For an expression in $\Lambda_\ell^{\mathrm{SFC}}$, Proposition 6.9 immediately yields an average variance of $\eta^{-\ell}$ (times a constant). For DSGD, we obtain an average variance of

$$\frac{1}{N} \cdot \sum_{k=1}^{N} \eta_k^{-\ell}$$

If $\eta_k \in \Theta(k^{-\frac{1}{\ell}+\epsilon})$ then (modulo constants), $\eta_k^{-\ell} = k^\delta$ for $\delta \leq 1$. Therefore, by Hölder's inequality,

$$\begin{aligned}
\frac{1}{N} \cdot \sum_{k=1}^{N} \eta_k^{-\ell} &= \frac{1}{N} \cdot \sum_{k=1}^{N} k^\delta \\
&\leq \frac{1}{N} \cdot \left(N^{1/\delta - 1} \cdot \sum_{k=1}^{N} k \right)^\delta \\
&= \frac{1}{N} \cdot \left(N^{1/\delta - 1} \cdot \frac{N \cdot (N+1)}{2} \right)^\delta \\
&= \left(\frac{N+1}{2} \right)^\delta = \eta_{\frac{N+1}{2}}^{-\ell}
\end{aligned}$$

As a consequence, the average variance of DSGD with initial η_0 is lower than for Fixed with coefficient η if $\eta < \eta_{\frac{N+1}{2}}$, the coefficient of DSGD after half the iterations.

6.4 Conclusion and Related Work

We have proposed a variant of SGD, *Diagonalisation Stochastic Gradient Descent*, and shown *provable* correctness. Our approach is based on our smoothed interpretation

of (possibly) discontinuous programs yielding unbiased gradient estimators. Crucially, asymptotically a stationary point of the original, *unsmoothed* problem is attained and a hyperparameter (accuracy of approximations) is tuned automatically. The correctness hinges on a careful analysis of the variance and a compatible scheme governing the accuracies. Notably, this purely depends on the (syntactic) structure of the program.

Related Work

Abstractly, our diagonalisation approach resembles graduated optimisation [Blake and Zisserman, 1987, Hazan et al., 2016]: a "hard" problem is solved by "simpler" approximations in such a way that the quality of approximation improves over time. However, the goals and merits of the approaches are incomparable: graduated optimisation is concerned with overcoming non-convexity of the objective function to find global optima rather than stationary points, whereas our approach is motivated by overcoming the bias of the gradient estimator of the objective function.

[Zang, 1981, Thm. 3.1] presents a smoothing approach for *non-stochastic* optimisation, which also enhances the approximation over time. However, they assume an *oracle* for *global optimisers* of each smoothed problem, which is an extremely strong assumption and renders their method impractical.

Chapter 7

Empirical Evaluation

At the end of the preceding chapter we have briefly analysed the variances of the methods proposed in this thesis—SGD with the reparameterisation gradient estimator for a fixed smoothing (FIXED) and DSGD. In this chapter, we conduct an empirical evaluation against our key competitors: standard SGD with the (SCORE) estimator [Ranganath et al., 2014, Wingate and Weber, 2013, Minh and Gregor, 2014], the biased reparameterisation estimator (REPARAM) and the unbiased correction of it (LYY18) due to [Lee et al., 2018].

7.1 Models

We include the models from [Lee et al., 2018], an example from differential privacy [Davidson-Pilon, 2015], a neural network for which our main competitor, the estimator of [Lee et al., 2018], is *not* applicable, and a truncated random walk.

The models from [Lee et al., 2018] are as follows:

- `temperature` [Soudjani et al., 2017] models a controller keeping the temperature of a room within set bounds. The discontinuity arises from the discrete state of the controller, being either on or off, which disrupts the continuous state representing the temperature of the room. Given a set of noisy measurements of the room temperature, the goal is to infer the controller state at each of 21 time steps. The model has a 41-dimensional latent variable and 80 if-statements.

- `textmsg` [Davidson-Pilon, 2015] models daily text message rates, and the goal is to discover a change in the rate over the 74-day period of data given. The non-differentiability arises from the point at which the rate is modelled to change. The model has a 3-dimensional latent variable (the two rates and the point at which they change) and 37 if-statements.

- `influenza` [Shumway and Stoffer, 2005] models the US influenza mortality for 1969. In each month, the mortality rate depends on the dominant virus strain being of type 1 or type 2, producing a non-differentiability for each month. Given the mortality data, the goal is to infer the dominant virus strain in each month. The model has a 37-dimensional latent variable and 24 if-statements.

Additionally, we introduce the following models:

– `random-walk` models a random walk (similar to Fig. 1.1a) of bounded length. The goal is to infer the starting position based on the distance walked. The walk stops as soon as the destination is reached. This is checked using if-statements and causes discontinuities. In each step a normal-distributed step is sampled and its absolute value is added to the distance walked so far, which accounts for more non-differentiabilities. Overall, the model has a 16-dimensional latent variable and 31 if-statements.

– `cheating` [Davidson-Pilon, 2015] simulates a differential privacy setting where students taking an exam are surveyed to determine the prevalence of cheating without exposing the details for any individual. Students are tasked to toss a coin, on heads they tell the truth (cheating or not cheating) and on tails they toss a second coin to determine their answer. The tossing of coins here is a source of discontinuity. The goal, given the proportion of students who answered yes, is to predict a posterior on the cheating rate. In this model there are 300 if-statements and a 301-dimensional latent space, although we only optimise over a single dimension with the other 300 being sources of randomness.

– `xornet` is a simple multi-layer neural network trained to compute the exclusive-or (XOR) function. It has a `2-4-2-1` network architecture with two inputs and one output, and all activation functions being the Heaviside step function which is traditionally infeasible for gradient-based optimisation because of the discontinuity at 0 and a zero gradient everywhere else. The model has a 25-dimensional latent space (for all the weights and biases) and 28 if-statements. Note that this model is not applicable to the LYY18 estimator since the branch conditions are not all affine in the latent space.

F_2 from Example 4.13 can be viewed as a (stark) simplification of `xornet`. It is also the only model in which the guards of if-statements contain variables which in turn depend on branching. As such, `xornet` corresponds to a term in Λ_3^{SFC}, whereas all other models correspond to Λ_1^{SFC}.

7.2 Experimental Set-Up

The (Python) implementation is based on [Lee et al., 2018] and the Part C Project of Basim Khajwal at the University of Oxford [Khajwal, 2022]. We employ the `jax` library to provide automatic differentiation which is used to implement each of the above estimators for an arbitrary (probabilistic) program. The smoothed interpretation can be obtained automatically by (recursively) replacing conditionals $\mathbf{if}\ E_1 < 0\ \mathbf{then}\ E_2\ \mathbf{else}\ E_3$ with $\sigma_\eta(-E_1) \cdot E_2 + \sigma_\eta(E_1) \cdot E_3$ in a preprocessing step. (We avoid a potential blow-up by using an auxiliary variable for E_1.)

In view of Theorem 6.11, for DSGD we choose the accuracy coefficient schemes $\eta_k := \eta_0/\sqrt{k}$ for $\eta_0 > 0$; due to the nesting of guards we use $\eta_k := \eta_0 \cdot k^{-0.2}$ for `xornet`. We compare (using the same line style) DSGD for different choices of η_0 to Fixed using the fixed accuracy coefficient corresponding to η_{4000}.

To enable a fair comparison to [Lee et al., 2018], we follow their set-up and use the state-of-the-art stochastic optimiser `Adam`[1] with a step size of `0.001`, except for `xornet` for which we use `0.01`, for 10,000 iterations. For each iteration, we use 16 Monte Carlo samples from the chosen estimator to compute the gradient. As in [Lee et al., 2018], the LYY18 estimator does not compute the boundary surface term exactly, but estimates it using a single subsample.

For every 100 iterations, we take 1000 samples of the estimator to estimate the current ELBO value and the variance of the gradient. Since the gradient is a vector, the variance is taken in two ways: averaging the component-wise variances and the variance of the L2 norm.

We separately benchmark each estimator by computing the number of iterations each can complete in a fixed time budget; the computational cost of each estimator is then estimated to be the reciprocal of this number. This then allows us to compute a set of *work-normalised* variances [Botev and Ridder, 2017] for each estimator, which are the product of the computational cost and the variance[2].

7.3 Analysis of Results

We plot the ELBO trajectories in Fig. 7.1 and include data on the computational cost and *work-normalised* variance [Botev and Ridder, 2017] in Table 7.1.

Experimentally, the bias of REPARAM becomes evident in `temperature`, `xornet`, `random-walk` and `cheating`, converging to suboptimal values. We can also see from Table 7.1 that the SCORE estimator exhibits very high variance, especially for `temperature` for which it not only converges slowly but even yields a very poor final ELBO (cf. Fig. 7.1a).

The trajectories for the DSGD and FIXED estimator perform comparably to the LYY18 estimator. However, they attain orders of magnitude reduction in work-normalised variance (3 to 20,000 x). Besides, for `xornet` LYY18 is *not* applicable as there are non-affine conditions in if-statements and for `random-walk` its variance is even higher than SCORE.

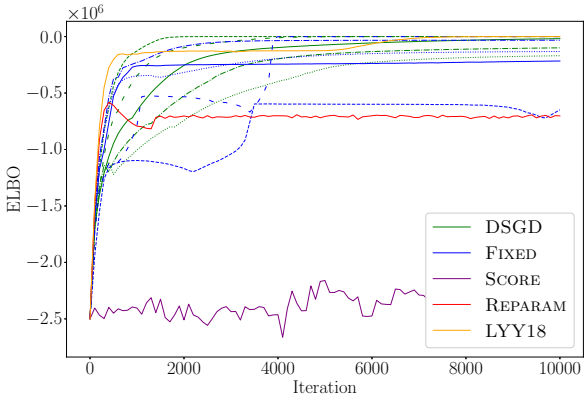
Compared to FIXED, we observe that DSGD is more robust[3] to the choice of (initial) accuracy coefficients (especially for `temperature` and `xornet`). Besides, there is a moderate improvement of variance (Table 7.1).

Summa summarum, the results confirm that the REPARAM estimator is biased and that the SMOOTH and DSGD estimators do not have the same limitation. Where the LYY18 estimator is defined, they converge to roughly the same objective value. Our smoothing approaches are applicable to more complex models, simpler and computationally more efficient (there is no need to compute a correction term). Besides, our estimators have consistently significantly lower work-normalised variance.

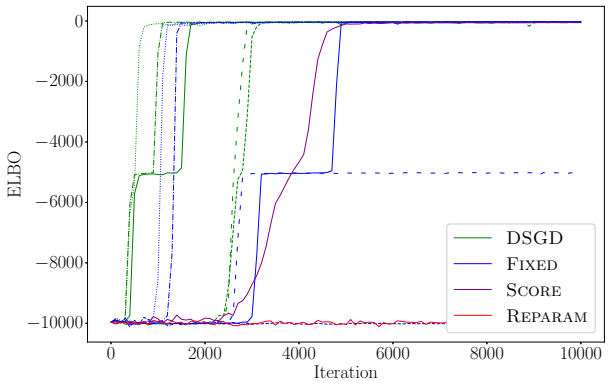
[1] together with the respective gradient estimators, e.g. $\nabla[\![F]\!]_{\eta_k}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))$ in step k for DSGD, resulting in a hybrid of DSGD and Adam

[2] This is a more suitable measure than "raw" variances since the latter can be improved routinely at the expense of computational efficiency by taking more samples (cf. discussion in Section 2.4).

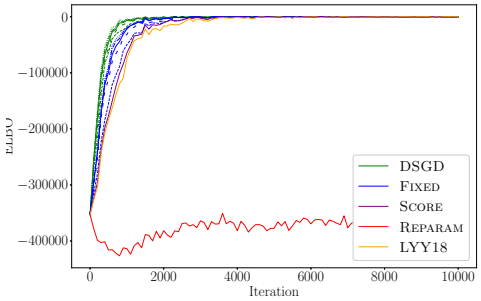
[3] On a *finite* run our *asymptotic* convergence result Theorem 6.11 cannot completely eliminate the dependence on this choice.



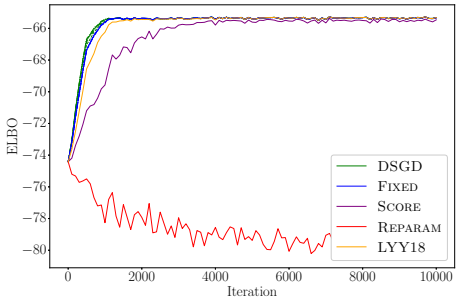
(a) `temperature`



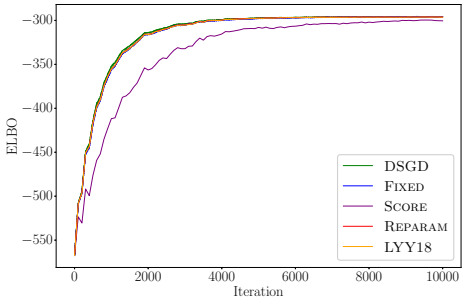
(b) `xornet`



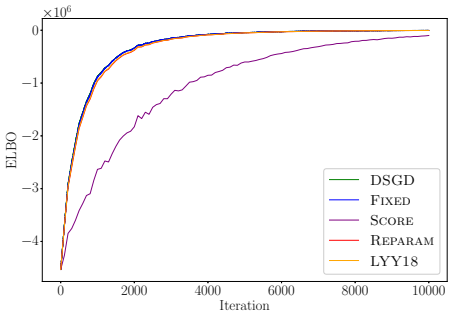
(c) `random-walk`



(d) `cheating`



(e) `textmsg`



(f) `influenza`

Figure 7.1: ELBO trajectories for each model. A single colour is used for each estimator and the choice of $\eta = \eta_{4000} = 0.06, 0.1, 0.14, 0.18, 0.22$ (which determines η_0) is represented by dashed, loosely dashed, solid, dash-dotted, dotted lines, respectively.

Table 7.1: Computational cost and work-normalised variances, all given as ratios with respect to the SCORE estimator (omitted since it would be all 1s). See Section 7.2 for details on how these metrics are obtained.

(a) `temperature`

Estimator	Cost	Avg($V(.)$)	$V(\|\|.\|\|_2)$
DSGD	1.71	4.91e-11	2.54e-10
FIXED	1.71	2.84e-10	2.24e-09
REPARAM	1.26	1.47e-08	1.94e-08
LYY18	9.61	1.05e-06	4.04e-05

(b) `xornet`

Estimator	Cost	Avg($V(.)$)	$V(\|\|.\|\|_2)$
DSGD	1.74	6.21e-03	3.66e-02
FIXED	1.87	1.21e-02	5.43e-02
REPARAM	0.388	8.34e-09	2.62e-09

(c) `random-walk`

Estimator	Cost	Avg($V(.)$)	$V(\|\|.\|\|_2)$
DSGD	4.70	1.71e-01	2.61e-01
FIXED	4.70	9.50e-01	1.49
REPARAM	2.17	8.63e-10	7.01e-10
LYY18	4.81	7.92	1.26e+01

(d) `cheating`

Estimator	Cost	Avg($V(.)$)	$V(\|\|.\|\|_2)$
DSGD	1.52	2.31e-03	3.51e-03
FIXED	1.52	2.84e-03	4.64e-03
REPARAM	9.36e-01	4.14e-19	1.16e-18
LYY18	2.59	4.27e-02	1.09e-01

(e) `textmsg`

Estimator	Cost	Avg($V(.)$)	$V(\|\|.\|\|_2)$
DSGD	1.84	7.89e-03	1.53e-02
FIXED	1.79	1.08e-02	2.14e-02
REPARAM	1.25	7.99e-03	1.53e-02
LYY18	4.51	3.42e-02	6.00e-02

(f) `influenza`

Estimator	Cost	Avg($V(.)$)	$V(\|\|.\|\|_2)$
DSGD	1.28	7.77e-03	3.94e-03
FIXED	1.28	7.92e-03	3.97e-03
REPARAM	1.21	7.60e-03	3.75e-03
LYY18	8.30	5.80e-02	2.88e-02

Chapter 8

Continuous but Non-Differentiable Models

Motivated by the fact that the reparameterisation gradient estimator may be biased for non-differentiable models, we have developed methods extending the approach to non-differentiable models. However, it is worth noting that the examples exhibiting the bias of the reparameterisation gradient estimator, Example 2.16 and Fig. 2.4, are discontinuous. So a natural question to investigate is the following:

> *Is the reparameterisation gradient estimator unbiased for non-differentiable but continuous models?*

In this chapter we answer this question affirmatively: under mild conditions the reparameterisation gradient estimator is unbiased for non-differentiable but continuous models. As a consequence, we can avoid taking smooth approximations and instead employ SGD with the reparameterisation gradient estimator directly on the unsmoothed objective function.

Furthermore, in Sections 8.2 to 8.5 we also present a method to verify continuity in the presence of conditionals based on higher-order logic.

The following examples illustrate that non-differentiable but continuous models are of practical interest:

Example 8.1. We model a temperature regulation system using a probabilistic program. Without intervention, the temperature fluctuates randomly. If the temperature drops below a threshold of 19 degrees centigrade the heating is engaged and the power is proportional to the deviation from the threshold. Time is discretised and after one time unit we measure a temperature of 21 degrees. We are interested in the distribution of the original temperature.

```
let t0 = sample normal(20,1)
    mu = t0 + if t0 < 19 then 2 * (19-t0)
                         else 0
    observe 21 from normal(mu, 1)
in t0
```

The (unnormalised) density of t_0 is plotted in Fig. 8.1, and we see that it is not differentiable (yet continuous) at $t_0 = 19$.

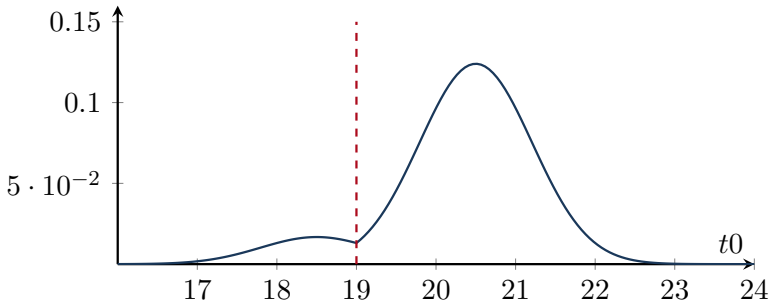


Figure 8.1: (Unnormalised) density for Example 8.1

Example 8.2. We model a one-dimensional random walk. Starting at some position near 0, we take 100 standard normal distributed steps. We observe that the total distance travelled, which is the absolute value of the sum of the steps, is 52, and we would like to infer the final position.

```
let p0    = sample 𝒩(0, σ0)   -- initial position
    s1    = sample 𝒩(0, 1)    -- sample steps
    ⋮
    s100  = sample 𝒩(0, 1)
    d     = |s1| + ⋯ + |s100| -- total distance travelled
    p100  = p0 + s1 + ⋯ + s100 -- final position
    observe 52 from 𝒩(d, σ)
in p100
```

where $\sigma_0, \sigma \in \mathbb{R}_{>0}$ are constants. Clearly, the density is continuous but non-differentiable.

8.1 Unbiasedness

First, we would like to abstractly show unbiasedness. To address this, it is worth revisiting the proof of Lemma 2.17 (which establishes unbiasedness).

Suppose $\theta_0 \in \mathbb{R}$ and $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$. Provided the integrals are finite,

$$\frac{\partial}{\partial \theta} \int f(\theta_0, \boldsymbol{s}) \, \mathrm{d}\boldsymbol{s} = \lim_{h \searrow 0} \int \frac{f(\theta_0 + h, \boldsymbol{s}) - f(\theta_0, \boldsymbol{s})}{h} \, \mathrm{d}\boldsymbol{s}$$

Now, we wish to exchange taking the limit with integration. This is valid by the Lebesgue dominated convergence theorem [Klenke, 2014] if $\frac{\partial f}{\partial \theta}(\theta_0, \boldsymbol{s})$ exists for almost all $\boldsymbol{s}$ and there exists an integrable $g : \mathbb{R}^n \to \mathbb{R}$ and $\delta > 0$ satisfying

$$\left| \frac{f(\theta_0 + h, \boldsymbol{s}) - f(\theta_0, \boldsymbol{s})}{h} \right| \leq g(\boldsymbol{s})$$

for all $\boldsymbol{s} \in \mathbb{R}^n$ and $0 < |h| < \delta$. Then

$$\lim_{h \searrow 0} \int \frac{f(\theta_0 + h, \boldsymbol{s}) - f(\theta_0, \boldsymbol{s})}{h} \, \mathrm{d}\boldsymbol{s} = \int \lim_{h \searrow 0} \frac{f(\theta_0 + h, \boldsymbol{s}) - f(\theta_0, \boldsymbol{s})}{h} \, \mathrm{d}\boldsymbol{s} = \int \frac{\partial f}{\partial \theta}(\theta_0, \boldsymbol{s}) \, \mathrm{d}\boldsymbol{s}$$

This proves:

Lemma 8.3. *Let $\boldsymbol{\Theta} \subseteq \mathbb{R}^m$ be open and convex. Suppose $f : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$ satisfies for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$,*

(i) *$\boldsymbol{s} \mapsto f(\boldsymbol{\theta}, \boldsymbol{s})$ is integrable*
(ii) *for almost all $\boldsymbol{s} \in \mathbb{R}^n$, $f(\boldsymbol{\theta}, \boldsymbol{s})$ is differentiable*
(iii) *there exists an integrable $g : \mathbb{R}^n \to \mathbb{R}$ and $\delta > 0$ satisfying*

$$\left| \frac{f(\boldsymbol{\theta} + \mathbf{e}_i, \boldsymbol{s}) - f(\boldsymbol{\theta}, \boldsymbol{s})}{h} \right| \leq g(\boldsymbol{s})$$

for all $0 < |h| < \delta$ and $\boldsymbol{s} \in \mathbb{R}^n$.

Then for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, $\frac{\partial}{\partial \theta_i} \int f(\boldsymbol{\theta}, \boldsymbol{s}) \, \mathrm{d}\boldsymbol{s} = \int \frac{\partial f}{\partial \theta_i}(\boldsymbol{\theta}, \boldsymbol{s}) \, \mathrm{d}\boldsymbol{s}$.

Note that the $\boldsymbol{s}$ where f is not differentiable may depend on the parameters $\boldsymbol{\theta}$.

Furthermore, in order to apply the dominated convergence theorem it is not necessary to assume that

$$\lim_{h \to 0} \left| \frac{f(\theta_0 + h, \boldsymbol{s}) - f(\theta_0, \boldsymbol{s})}{h} \right|$$

exists for *all* $\boldsymbol{s} \in \mathbb{R}^n$. If we *do* assume this and the existence of an integrable $g : \mathbb{R}^n \to \mathbb{R}$ satisfying

$$\left| \frac{\partial f}{\partial \theta}(\theta, \boldsymbol{s}) \right| \leq g(\boldsymbol{s})$$

for all $\theta \in \Theta$ and $\boldsymbol{s} \in \mathbb{R}$ (as in the third premise of Lemma 2.17) then by the mean value theorem for all $\theta \in \Theta$ and h such that $\theta + h \in \Theta$

$$\left| \frac{f(\theta + h, \boldsymbol{s}) - f(\theta, \boldsymbol{s})}{h} \right| = \left| \frac{\partial f}{\partial \theta}(\theta + \zeta, \boldsymbol{s}) \right| \leq g(\boldsymbol{s})$$

for some $0 < |\zeta| < |h|$.

8.1.1 Application to the Programming Language

Now, we wish to apply the insight to terms $\boldsymbol{\theta} \mid \Sigma \vdash_{\text{int}} M : R^{(\mathbf{p})}$ in our programming language. We modify Assumption 5.9 slightly[1]:

Assumption 8.4. $\mathrm{Op} = \mathrm{Op}_{\mathrm{pb}} \cup \{\exp, \log, (-)^{-1}\}$, where $\mathrm{Op}_{\mathrm{pb}}$ is a set of *analytic* functions bounded by polynomials, partial derivatives of which are bounded by polynomials.

Clearly, the first premise of Lemma 8.3 is satisfied by Proposition 4.26. Besides, analytic functions satisfy Assumption 3.1 (Example 3.2). Therefore, the second premise follows with our almost everywhere differentiability result (Theorem 3.18).

For the third, we note that we can easily obtain a variant of Lemmas 5.18 and 4.25: If $\boldsymbol{\theta} \mid \Sigma \vdash_{\text{int}} M : R$ then

$$[\![M]\!](\boldsymbol{\theta}, \boldsymbol{s}) = \sum_{i=1}^{\ell} [(\boldsymbol{\theta}, \boldsymbol{s}) \in U_i] \cdot f_i(\boldsymbol{\theta}, \boldsymbol{s})$$

[1] This will be required in Section 8.3 for proving decidability results.

where $U_1, \ldots, U_\ell$ is a partition of $\boldsymbol{\Theta} \times \mathbb{R}^n$ and the partial derivatives of $f_1, \ldots, f_\ell$ are uniformly bounded by polynomials.

If $[\![M]\!]$ is *continuous* then for any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, $0 < |h| < 1$ and $\boldsymbol{s}$,

$$\left| \frac{[\![M]\!](\boldsymbol{\theta} + h \cdot \mathbf{e}_i, \boldsymbol{s}) - [\![M]\!](\boldsymbol{\theta}, \boldsymbol{s})}{h} \right| \leq \sum_{j=1}^{\ell} p_j(\boldsymbol{s})$$

where $p_1, \ldots, p_\ell$ are uniform polynomial bound on $\frac{\partial f_1}{\partial \theta_i}, \ldots, \frac{\partial f_\ell}{\partial \theta_i}$, respectively (see Lemma D.1 in Appendix D for a proof). Furthermore, since $\boldsymbol{s} \mapsto \boldsymbol{\mathcal{D}}(\boldsymbol{s}) \cdot p_i(\boldsymbol{s})$ is integrable, we conclude with Lemma 8.3:

Theorem 8.5 (Unbiasedness)**.** *If* $\boldsymbol{\theta} \mid \Sigma \vdash_{\text{int}} M : R$ *and* $[\![M]\!]$ *is continuous then*

$$\nabla_{\boldsymbol{\theta}} \, \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}}[[\![M]\!](\boldsymbol{\theta}, \boldsymbol{s})] = \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}}[\nabla_{\boldsymbol{\theta}} \, [\![M]\!](\boldsymbol{\theta}, \boldsymbol{s})]$$

A simple way to guarantee continuity is to ban conditionals altogether. However, we can also allow a stock set COp of terms $\mid [] : M : \iota_1^{(\mathbf{p})} \to \cdots \to \iota_k^{(\mathbf{p})} \to \iota^{(\mathbf{p})}$ for which $[\![M]\!]$ is continuous:

$$C ::= x \mid \underline{f}(C, \ldots, C) \mid \lambda y.\, C \mid C\, C \mid \mathbf{sample}_{\mathcal{D}} \rhd \varphi_{\boldsymbol{\theta}} \mid M$$

provided $M \in \text{COp}$. Morally, this corresponds to allowing continuous but non-differentiable primitives.

Lemma 8.6. *Let* $\boldsymbol{\theta} \mid \Sigma \vdash_{\text{tr}} C : \iota$*. If* $[\![M]\!]$ *is continuous for all* $M \in \text{COp}$ *then* $[\![C]\!]$ *is continuous.*

For instance, we can admit the following continuous but non-differentiable terms to COp:

$$\begin{aligned} \text{ReLU} &\equiv \lambda x.\, \mathbf{if}\, x < 0\, \mathbf{then}\, 0\, \mathbf{else}\, x \\ \text{abs} &\equiv \lambda x.\, \mathbf{if}\, x < 0\, \mathbf{then}\, -x\, \mathbf{else}\, x \\ \max &\equiv \lambda x, y.\, \mathbf{if}\, x - y < 0\, \mathbf{then}\, y\, \mathbf{else}\, x \end{aligned}$$

Thus, we can re-write the conditional $t_0 + \mathbf{if}\, t_0 - 19 < 0\, \mathbf{then}\, 2 \cdot (19 - t_0)\, \mathbf{else}\, 0$ in Example 8.1 as $t_0 + 2 \cdot \text{ReLU}\,(19 - t_0)$.

8.1.2 Continuity in the Presence of Conditionals

Are there syntactic methods to establish continuity for terms with conditionals? In earlier chapters we employed various type systems to verify properties, which are pre-conditions of inference algorithms. One of the features of type systems is compositionality, which is usually considered a strength because it enables efficient and simple verification. However, the type systems we have developed thus far are incomplete (cf. e.g. Example 5.39) because the (arithmetic) properties we have investigated do not have a (fully) compositional nature.

Example 8.7. Consider the following terms in the simple function calculus:

$$F \equiv \mathbf{if}\, x - y < 0\, \mathbf{then}\, F_1\, \mathbf{else}\, F_2$$

$$F_1 \equiv \textbf{if } x < 0 \textbf{ then } 0 \textbf{ else } 2x$$
$$F_2 \equiv \textbf{if } y < 0 \textbf{ then } x \textbf{ else } x + y$$

Note that both F_1 and F_2 are continuous. However, F is not continuous. (For $x_k := -1-1/k$ and $y_k := -1$, $\lim[\![F]\!](x_k, y_k) = 0$ but $[\![F]\!](\lim x_k, \lim y_k) = [\![F]\!](-1,-1) = -1$.)

A possible method to detect this would be to check that for the boundary $x - y = 0$ of the outer conditional all branches (i.e. 0, $2x$, x, $x + y$) agree.

However, this method unnecessarily rules out continuous terms:

Example 8.8. The term $\textbf{if } x+1 < 0 \textbf{ then } -x \textbf{ else } (\textbf{if } x-1 < 0 \textbf{ then } 1 \textbf{ else } x)$ implements $x \mapsto \max\{|x|, 1\}$. For $x + 1 = 0$ the branches $-x$ and x do not agree.

These examples suggest that even for the case of the simple function calculus, purely syntactic methods based on type systems are highly incomplete and not truly compositional.

In the remainder of this chapter we hence pursue a different approach:

(i) We propose a higher-order logic with a non-standard background theory and prove satisfiability to be decidable.
(ii) We then provide a sound but incomplete encoding of continuity.
(iii) Finally, we introduce a novel predicate with non-standard semantics for which an encoding is not only sound but also complete, and we prove a decidability result.

8.2 Higher-Order Constrained Clauses

Constrained Horn clauses, a fragment of first-order logic with respect to background theories, are a very successful approach for program safety verification [Bjørner et al., 2015, Bjørner et al., 2012, Grebenshchikov et al., 2012]. They are conceptually appealing because they help *separate specifications from their verification* [Aiken, 1999]. Current state-of-the-art tools leverage the power of SAT and SMT-solvers (see e.g. [Barrett et al., 2021]), which have seen enormous progress in the past three decades, and can be employed to verify code arising in industry with several thousand lines of code.

More recently, [Cathcart Burn et al., 2018] proposed a higher-order extension—*higher-order constrained Horn clauses* (or *HoCHC* for short)—which is suitable for the verification of functional programs. To demonstrate that a program is *safe*, it suffices to find overapproximations of the input-output-graph (i.e. *invariants*) of the functions that imply the required property. The idea then is to express the problem of finding such an overapproximation *logically* as a satisfiability problem for a higher-order constrained system.

The following example illustrates that higher-order logic is a useful framework to capture higher-order, functional computation directly:

Example 8.9. Consider the term:

```
let branch x y = if x < 0 then y else x
    twice f x  = f (f x)
in  twice (branch sample) sample
```

for which we wish to axiomatise over-approximations of its input-output-graph. We introduce relational[2] symbols $\text{Branch} : \iota \to \iota \to \iota \to o$, $\text{Twice} : (\iota \to \iota \to o) \to \iota \to \iota \to o$ and $\text{Term} : \iota \to \iota \to \iota \to o$ for the graphs of branch, twice and the overall term, respectively. We axiomatise their graphs as follows:

$$\begin{aligned} x < 0 &\to \text{Branch}\, x\, y\, y \\ x \geq 0 &\to \text{Branch}\, x\, y\, x \\ p\, x\, y \wedge p\, y\, z &\to \text{Twice}\, p\, x\, z \\ \text{Twice}\,(\text{Branch}\, s_1)\, s_2\, z &\to \text{Term}\, s_1\, s_2\, z \end{aligned}$$

where $x, y, z : \iota$, $p : \iota \to \iota \to o$. Besides, we could express the (incorrect) safety conjecture that the term never returns 0 via

$$\text{Term}\, s_1\, s_2\, y \wedge y = 0 \to \bot$$

[Ong and Wagner, 2019] study the algorithmic, model-theoretic and semantical properties of higher-order Horn clauses with a 1st-order background theory. They show that it is a *refutationally complete and semantically invariant* system of higher-order logic modulo theories. As such, it occupies a "sweet spot" in higher-order logic.

[Ong and Wagner, 2019] propose a simple proof system, which combines 1st-order reasoning in the background theory with purely logical reasoning agnostic of the background theory via a higher-order variant of resolution. The proof system assumes an oracle for the satisfiability of conjunctions of constraints in the background theory. They prove soundness and completeness: *unsatisfiability* can be proven *syntactically*. This result does not directly provide a method for showing *satisfiability*. Indeed, satisfiability of HoCHC is undecidable in general [Ong and Wagner, 2019]. An intuitive reason for this is that (unrestricted) HoCHC can be used to encode recursion. Finally, [Ong and Wagner, 2019, Cathcart Burn et al., 2021, Jochems et al., 2023] study decidable fragments.

In the remainder of this section we give a more formal account of higher-order constrained Horn clauses tailored to our intended application: the verification of continuity.

8.2.1 Background Theory

In the light of the optimisation framework of Chapter 4, we may be tempted to choose the theory of reals and smooth functions with equalities and inequalities as our background theory. Whilst this would naturally accommodate our programming language, it would however mean that for deciding satisfiability of higher-order constrained Horn clauses we would need to decide satisfiability of conjunctions in the background theory such as

$$f_1(\boldsymbol{x}) \bowtie_1 g_1(\boldsymbol{x}) \wedge \cdots \wedge f_k(\boldsymbol{x}) \bowtie_k g_k(\boldsymbol{x})$$

where $f_1, g_1, \ldots, f_k, g_k$ are smooth functions and $\bowtie_1, \ldots, \bowtie_k \in \{=, <, \leq, \neq\}$. On the other hand, *linear real arithmetic*, where the functions are affine, is very well studied and can be efficiently decided (both from a practical and (complexity) theoretic perspective). However, this would mean that we could only allow addition and multiplication with constants as our primitive operations Op, which would restrict the expressivity enormously. In particular, we would not be able to express density functions.

[2] o is the sort of Booleans

We choose a compromise: dis-equations ($\neq$) can use arbitrary analytic functions whilst (in-)equalities ($<, \leq, =$) must be affine. Intuitively, this will force us to only use affine terms in guards, which is a mild restriction (cf. Fig. 8.4).

8.2.2 Syntax

Formally, we modify our logic to have four ***base sorts*** $\iota^{(a)}$, where ι is R or $R_{>0}$ and $a \in \{\mathbf{t}, \mathbf{f}\}$ denotes whether the expression is affine ($\mathbf{t}$) or just analytic ($\mathbf{f}$), and we use the subsorting relation of Fig. 8.2a. This is motivated by the fact that every affine term is also analytic. We intend to interpret $[\![R^{(a)}]\!] \coloneqq \mathbb{R}$ and $[\![R^{(a)}_{>0}]\!] \coloneqq \mathbb{R}_{>0}$.

Besides, ***relational*** and ***argument sorts*** are generated from

base sorts	$\iota ::= R^{(\mathbf{f})} \mid R^{(\mathbf{t})} \mid R^{(\mathbf{f})}_{>0} \mid R^{(\mathbf{t})}_{>0}$
relational sorts	$\rho ::= o \mid \sigma \to \rho$
argument sorts	$\sigma ::= \iota \mid \rho$

where o is the sort of truth values.

We assume a fixed supply Rel of (sorted) relational symbols $P_1 : \rho_1, \ldots$ and analytic primitive operations Op (incl. constants). Contexts (typically denoted by Δ) are finite lists of sorted variables $x_1 : \sigma_1, \ldots, x_n : \sigma_n$. Sorting judgements for terms (typically denoted by t and variants thereof) are standard (see Fig. 8.2b). Note that terms $\Delta \vdash t : \iota^{(a)}$ cannot contain relational symbols and only variables of type $x_i : \iota_i^{(a_i)}$.

An ***atom*** is a term of sort o. A ***background atom*** has the form $t_1 \bowtie t_2$, where $\bowtie \in \{=, \leq, <, \neq\}$, and is typically denoted by φ (and variants thereof). An atom which is not a background atom is sometimes referred to as ***foreground atom***.

Definition 8.10. A ***body*** has the form

$$\varphi_1 \wedge \cdots \wedge \varphi_\ell \wedge t_1\, y_1 \wedge \cdots \wedge t_m\, y_m$$

where for some context Δ,

(i) $\Delta \vdash \varphi_1 : o, \ldots, \Delta \vdash \varphi_\ell : o$ are background atoms
(ii) $\Delta \vdash t_1\, y_1 : o, \ldots, \Delta \vdash t_m\, y_m : o$ are foreground atoms such that for each $1 \leq i \leq m$, $y_i : \iota_i^{(a_i)} \in \Delta$ is a variable not occurring in $t_1, \ldots, t_i$
(iii) the variables $y_1, \ldots, y_m$ are *pairwise distinct*.

We typically denote bodies by B or B' and use $\top$ for the empty conjunction.

There are two kinds of ***higher-order Constrained Horn Clauses (HoCHCs)***:

Definition 8.11. A ***goal clause*** has the form $B \to \bot$, where B is a body in which all variables have base sort.

Definition 8.12. A ***definite clause*** has the form

$$\varphi_1 \wedge \cdots \wedge \varphi_\ell \wedge t_1\, y_1 \wedge \cdots \wedge t_m\, y_m \to P\, x_1 \cdots x_n\, t$$

where for some context Δ,

(i) $\varphi_1 \wedge \cdots \wedge \varphi_\ell \wedge t_1\, y_1 \wedge \cdots \wedge t_m\, y_m$ is a body

$$\frac{}{\iota \sqsubseteq \iota} \qquad \frac{}{R^{(\mathbf{t})} \sqsubseteq R^{(\mathbf{f})}} \qquad \frac{}{R^{(\mathbf{t})}_{>0} \sqsubseteq R^{(\mathbf{f})}_{>0}}$$

$$\frac{}{R^{(\mathbf{f})}_{>0} \sqsubseteq R^{(\mathbf{f})}} \qquad \frac{}{R^{(\mathbf{t})}_{>0} \sqsubseteq R^{(\mathbf{t})}} \qquad \frac{}{R^{(\mathbf{t})}_{>0} \sqsubseteq R^{(\mathbf{f})}}$$

$$\frac{\sigma' \sqsubseteq \sigma \quad \rho \sqsubseteq \rho'}{(\sigma \to \rho) \sqsubseteq (\sigma' \to \rho')}$$

(a) Subsorting

$$\frac{\Delta \vdash t : \sigma}{\Delta \vdash t : \sigma'}\ \sigma \sqsubseteq \sigma'$$

$$\frac{}{\Delta \vdash P : \rho}\ P : \rho \in \mathrm{Rel} \qquad \frac{}{\Delta \vdash x : \sigma}\ x : \sigma \in \Delta$$

$$\frac{\Delta \vdash t : \sigma \to \rho \quad \Delta \vdash t' : \sigma}{\Delta \vdash t\, t' : \rho}$$

$$\frac{t_1 : \iota_1^{(\mathbf{f})} \quad \cdots \quad t_\ell : \iota_\ell^{(\mathbf{f})}}{\underline{f}(t_1, \ldots, t_\ell) : \iota^{(\mathbf{f})}}\ f : [\![\iota_1]\!] \times \cdots \times [\![\iota_\ell]\!] \to [\![\iota]\!] \in \mathrm{Op} \qquad \frac{t_1 : \iota^{(\mathbf{t})} \quad t_2 : \iota^{(\mathbf{t})}}{c \cdot t_1 + t_2 : \iota^{(\mathbf{t})}}\ c \in \mathbb{R}$$

$$\frac{t_1 : R^{(\mathbf{t})} \quad t_2 : R^{(\mathbf{t})}}{t_1 \bowtie t_2 : o}\ \bowtie \in \{=, \leq, <\} \qquad \frac{t_1 : R^{(\mathbf{f})} \quad t_2 : R^{(\mathbf{f})}}{t_1 \neq t_2 : o}$$

(b) Sorting rules for terms

Figure 8.2: Sorting for higher-order constrained Horn clauses

(ii) $\Delta \vdash P\, x_1 \cdots x_n\, t : o$ and $\Delta \vdash t : \iota^{(a)}$
(iii) the variables $y_1, \ldots, y_m, x_1, \ldots, x_n$ are *pairwise distinct*
(iv) all non-base sort variables are amongst the x_i.

Example 8.13. (i) Example 8.9 presents higher-order definite clauses.
(ii) The following are 1st-order clauses (cf. Example 8.32):

$$\begin{aligned} z \leq 0 &\to P\, z\, (-z^2 + z - 1) \\ z \geq 0 &\to P\, z\, (z + 1) \\ P\, z\, y \wedge P\, z\, y' \wedge y \neq y' &\to \bot \end{aligned}$$

Remark 8.14. This definition of clauses is superficially different from [Ong and Wagner, 2019], where definite clauses are presented in the following disjunctive form

$$\neg A_1 \vee \cdots \vee \neg A_m \vee P\, x_1 \cdots x_{n+1}$$

where $x_1, \ldots, x_{n+1}$ are distinct variables and the atoms $A_1, \ldots, A_m$ are unrestricted (but the final argument in the head is a variable). Note that we can frame a definite clause $B \to P\, x_1 \cdots x_n\, M$, where $M : \iota$ easily in this form by using an auxiliary variable: $B \wedge x_{n+1} = M \to P\, x_1 \cdots x_n\, x_{n+1}$.

$$[\![R^{(a)}]\!] \coloneqq \mathbb{R} \qquad [\![R^{(a)}_{>0}]\!] \coloneqq \mathbb{R}_{>0} \qquad [\![o]\!] \coloneqq \{0,1\}$$
$$[\![\sigma \to \rho]\!] = [[\![\sigma]\!] \to [\![\rho]\!]]$$

(a) Interpretation of sorts

$$\begin{aligned}
\mathcal{A}[\![P]\!](\alpha) &\coloneqq P^{\mathcal{A}} \\
\mathcal{A}[\![x]\!](\alpha) &\coloneqq \alpha(x) \\
\mathcal{A}[\![t\,t']\!](\alpha) &\coloneqq \mathcal{A}[\![t]\!](\alpha)(\mathcal{A}[\![t']\!](\alpha)) \\
\mathcal{A}[\![\underline{f}(t_1,\ldots,t_\ell)]\!](\alpha) &\coloneqq f(\mathcal{A}[\![t_1]\!](\alpha),\ldots,\mathcal{A}[\![t_\ell]\!](\alpha)) \\
\mathcal{A}[\![c\cdot t_1 + t_2]\!](\alpha) &\coloneqq c\cdot\mathcal{A}[\![t_1]\!](\alpha) + \mathcal{A}[\![t_2]\!](\alpha) \\
\mathcal{A}[\![t_1 \bowtie t_2]\!](\alpha) &\coloneqq \begin{cases} 1 & \text{if } \mathcal{A}[\![t_1]\!](\alpha) \bowtie \mathcal{A}[\![t_2]\!](\alpha) \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

(b) Interpretation of terms

Figure 8.3: Interpretation of sorts and terms

Our modified definition has an important technical advantage: our non-standard[3] background theory does not need to include equality of analytic terms. (In fact the decision procedure of Section 8.3.1 cannot easily be extended to include equations in addition to inequations of analytic terms.) Besides, the encodings Fig. 8.5 and Eq. (8.6) will be arguably more natural.

8.2.3 Semantics

For higher-order logics there are multiple variants of semantics, which differ in their interpretation of higher sorts. In *standard semantics* they are interpreted as full (set-theoretic) function spaces. Unfortunately, full higher-order logic in standard semantics is wildly undecidable. E.g. the set[4] $\mathbf{V}^2(=)$ of valid sentences of the 2nd-order language of equality is not even analytical [Enderton, 2001].

For *Henkin semantics* [Henkin, 1950], which is, after all, "nothing but many-sorted 1st-order logic with comprehension axioms" [Enderton, 2001] (see also [Benthem and Doets, 1983, Leivant, 1994]), semi-decidability is recovered.

Remarkably, satisfiability of higher-order constrained Horn clauses does *not* depend on the choice of semantics[5] [Ong and Wagner, 2019].

Therefore, we employ the *standard* semantics, which interprets higher sorts as (full) function spaces (see Fig. 8.3a). A ***structure*** $\mathcal{A}$ assigns to each relational symbol $P : \rho \in \mathrm{Rel}$ an element of $P^{\mathcal{A}} \in [\![\rho]\!]$. A ***valuation*** α for a context Δ assigns to each $x : \sigma \in \Delta$, $\alpha(x) \in [\![\sigma]\!]$. Given a structure $\mathcal{A}$ and a valuation α, a term $\Delta \vdash t : \sigma$ is interpreted

[3] in the sense that dis-equations ($\neq$) can use arbitrary analytic functions whilst (in-)equalities ($<, \leq, =$) must be affine

[4] Define $\mathbf{V}^n(P)$ to be the set of valid sentences of nth-order logic with 2-place predicate P. Then $\mathbf{V}^1(=)$ is recursively enumerable.

[5] within the reasonable bounds formalised by *(complete) frames*

as $\mathcal{A}[\![t]\!](\alpha) \in [\![\sigma]\!]$ (see Fig. 8.3b). For an atom A we write $\mathcal{A}, \alpha \models A$ iff $\mathcal{A}[\![A]\!](\alpha) = 1$. Since terms $\Delta \vdash t : \iota$ and $\Delta \vdash t' : \iota$ do not contain relational symbols we abbreviate $[\![t]\!](\alpha) \coloneqq \mathcal{A}[\![t]\!](\alpha)$ and $\alpha \models t \bowtie t'$ iff $\mathcal{A}, \alpha \models t \bowtie t'$.

For a definite clause $\mathcal{A} \models A_1 \wedge \cdots \wedge A_m \to P\, x_1 \cdots x_n\, t$ if for all valuations α, $\mathcal{A}, \alpha \models A_1, \ldots, \mathcal{A}, \alpha \models A_m$ implies $\mathcal{A}, \alpha \models P\, x_1 \cdots x_n\, t$. Likewise, for a goal clause $A_1 \wedge \cdots \wedge A_m \to \bot$, we write $\mathcal{A} \models A_1 \wedge \cdots \wedge A_m \to \bot$ if for all valuations α, $\mathcal{A}, \alpha \not\models A_i$ for some $1 \leq i \leq m$.

A set Φ of clauses is ***satisfiable*** if there exists a structure $\mathcal{A}$ such that for all $C \in \Phi$, $\mathcal{A} \models C$.

8.2.4 Proof System

We adapt the resolution proof system of [Ong and Wagner, 2019] to our setting[6]:

Resolution
$$\frac{B \to P\, x_1 \cdots x_n\, t' \qquad P\, t_1 \cdots t_n\, y \wedge B' \to \bot}{(B \wedge B'[t'/y])[t_1/x_1, \ldots, t_n/x_n] \to \bot}$$

Constraint Refutation
$$\frac{\varphi_1 \wedge \cdots \wedge \varphi_\ell \to \bot}{\bot}$$

provided there exists a valuation α satisfying $\alpha \models \varphi_1 \wedge \cdots \wedge \varphi_\ell$.

In the resolution rule we assume that the variables have been renamed so that the clauses do not share variables. We write $\Phi \vdash_{\text{res}} G$ if G can be derived by the rules of the proof system from clauses in Φ.

Example 8.15. The clauses Φ in Example 8.13(ii) can be refuted as follows:

$$\dfrac{\dfrac{z \leq 0 \to P\, z\, (-z^2 + z - 1) \qquad P\, z\, y \wedge P\, z\, y' \wedge y \neq y' \to \bot}{z \leq 0 \wedge P\, z\, y' \wedge (-z^2 + z - 1 \neq y')} \qquad z \geq 0 \to P\, z\, (z+1)}{\dfrac{z \leq 0 \wedge (z \geq 0) \wedge (-z^2 + z - 1 \neq z + 1) \to \bot}{\bot}}$$

In the last step the constraint refutation rule is applicable because for $\alpha(z) \coloneqq 0$, $\alpha \models z \leq 0 \wedge (z \geq 0) \wedge (-z^2 + z - 1 \neq z + 1)$.

Soundness and completeness carry over to the modified framework:

Theorem 8.16 (Soundness and Completeness [Ong and Wagner, 2019])**.** *Let Φ be a set of higher-order constrained Horn clauses. Then Φ is unsatisfiable iff $\Phi \vdash_{\text{res}} \bot$.*

For soundness in the modified setting it is important to note that

$$(P\, x_1 \cdots x_n\, t')[\boldsymbol{t}/\boldsymbol{x}] \equiv P\, t_1 \cdots t_n\, t'' \equiv ((P\, t_1 \cdots t_n\, y)[t'/y])[\boldsymbol{t}/\boldsymbol{x}] \tag{8.1}$$

for $t'' \equiv t'[\boldsymbol{t}/\boldsymbol{x}]$ because y and x_i do not occur in $t_1, \ldots, t_n$ (the former by definition, the latter because variables have been renamed). Besides, the resolvent of a definite clause and a goal clause is a goal clause (according to Definition 8.11, see Lemma D.2).

The following example illustrates difficulties which would arise if we allowed variables to occur in multiple trailing positions:

[6] In particular, there is no need for their β-reduction rule.

Example 8.17. Consider the unsatisfiable set of clauses

$$\top \to P\,(x^2) \qquad \top \to Q\,(\exp(y)) \qquad P\,z \wedge Q\,z \to \bot$$

However, applying the resolution rule once, we obtain

$$\frac{\top \to Q\,(\exp(y)) \quad P\,z \wedge Q\,z \to \bot}{P\,(\exp(y)) \to \bot}$$

and we cannot proceed further. Indeed, if we tried to "unify" $\exp(y)$ and x^2 we would need to include arithmetic reasoning. One of our key design principles is the strict separation of purely logical reasoning and reasoning in the background theory.

8.3 Decidability of Recursion-Free HoCHCs

The objective of this section is to show that satisfiability of a fragment of higher-order constrained Horn clauses with our background theory suitable for the verification of continuity is decidable.

8.3.1 Decidability of the Background Theory

First, we need to provide a decision procedure for conjunctions of background atoms to determine the applicability of the constraint refutation rule. Since affine and analytic functions are closed under composition and subtraction, a conjunction of background atoms can be written compactly as:

$$\underbrace{A\boldsymbol{x} \bowtie \boldsymbol{b}}_{\text{affine}} \wedge \underbrace{\boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0}}_{\text{analytic}} \to \bot$$

where $\boldsymbol{x} = (x_1, \ldots, x_n)^T$, $A \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{f} = (f_1, \ldots, f_k) : \mathbb{R}^n \to \mathbb{R}^k$ are analytic, and $\bowtie$ includes $=, \leq$ and $<$ (in each row). Slightly unusually, by $\boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0}$ in Eq. (8.2) we mean $f_1(\boldsymbol{x}) \neq 0 \wedge \cdots \wedge f_k(\boldsymbol{x}) \neq 0$.

Thus, in this subsection we wish to develop decision procedures for

$$\models \underbrace{A\boldsymbol{x} \bowtie \boldsymbol{b}}_{\text{affine}} \wedge \underbrace{\boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0}}_{\text{analytic}} \to \bot \tag{8.2}$$

Recall the dichotomy:

Lemma 2.31. *If $f : \mathbb{R}^n \to \mathbb{R}$ is analytic then $f = 0$ (constantly) or $f \neq 0$ almost everywhere.*

We harness this insight in the extremely simple randomised Algorithm 1 for checking the special case

$$\models \boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0} \to \bot \tag{8.3}$$

by evaluating a single sample. If Eq. (8.3) holds, then clearly for *every* sample $\boldsymbol{s}$, $f_i(\boldsymbol{s}) = 0$ for some $1 \leq i \leq k$. Conversely, if Eq. (8.3) fails then $f_1, \ldots, f_k$ are not constantly 0 and therefore almost everywhere not 0 by Lemma 2.31. Thus, with probability 1, $f_i(\boldsymbol{s}) \neq 0$ for some $1 \leq i \leq k$, where $\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{D}}$ is a distribution on $\mathbb{R}^n$, which is absolutely continuous w.r.t. the Lebesgue measure (for instance the multivariate standard normal distribution). Consequently:

Data: $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^k$ are analytic
Result: $\models \boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0} \to \bot$
sample $\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}$;
if $f_i(\boldsymbol{s}) = 0$ *for some* i **then**
  return true;
else
  return false;
end

Algorithm 1: Randomised decision procedure for Eq. (8.3)

Lemma 8.18. *With probability 1,* $\models \boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0} \to \bot$ *iff Algorithm 1 returns true.*

Next, note that checking the satisfiability of

$$A\boldsymbol{x} \bowtie \boldsymbol{b}$$

amounts to checking the non-emptiness of a polytope or equivalently the satisfiability of a conjunction of constraints in the theory of linear real arithmetic (LRA) for which very efficient off-the-shelf solvers exist (and there are polynomial time complexity results) [Barrett et al., 2021, Kroening and Strichman, 2008].

If $A\boldsymbol{x} \bowtie \boldsymbol{b}$ is unsatisfiable then Eq. (8.2) clearly holds.

Next, to approach mixing linear constraints and affine inequations as in Eq. (8.2) when $A\boldsymbol{x} \bowtie \boldsymbol{b}$ is satisfiable, we start by considering the special case

$$\models A\boldsymbol{x} = \boldsymbol{b} \wedge \boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0} \to \bot \tag{8.4}$$

For $U \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} = \boldsymbol{b}\} \neq \boldsymbol{0}$, clearly Eq. (8.4) is valid iff $f_{|U} = \boldsymbol{0}$.

Let $r \leq m, n$ be the rank of A. Note that $U = \{C\boldsymbol{z} + \boldsymbol{d} \mid \boldsymbol{z} \in \mathbb{R}^{n-r}\}$ for some $C \in \mathbb{R}^{n \times (n-r)}$ and $\boldsymbol{d} \in \mathbb{R}^n$, which can be obtained by Gaussian elimination.

Example 8.19. Suppose

$$\begin{aligned} 2x_1 + x_2 + x_3 &= 1 \\ x_1 - x_2 &= 0 \\ 3x_2 + x_3 &= 1 \end{aligned}$$

which has rank 2. Then

$$U = \{(1, 1, -3)^T \cdot z + (0, 0, 1)^T \mid z \in \mathbb{R}\}.$$

Consequently, Eq. (8.4) holds iff

$$\underbrace{f(\boldsymbol{h}(\boldsymbol{z}))}_{\text{analytic}} \neq 0$$

for all $\boldsymbol{z} \in \mathbb{R}^{n-r}$, where $\boldsymbol{h}(\boldsymbol{z}) \coloneqq C\boldsymbol{z} + \boldsymbol{d}$. Crucially, Algorithm 1 can be applied to $f \circ \boldsymbol{h}$, yielding a randomised check of Eq. (8.4).

Next, we wish to decide the general case:

$$\models A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{f}(\boldsymbol{x}) \neq 0 \to \bot \tag{8.5}$$

The following is a key insight for a reduction to the special case just including equalities:

Lemma 8.20. *Suppose* $A\boldsymbol{x} \bowtie \boldsymbol{b}$ *and* $\boldsymbol{a}^T\boldsymbol{x} < b'$ *are (jointly) satisfiable and the following holds* $\models A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{a}^T\boldsymbol{x} < b' \wedge \boldsymbol{f}(\boldsymbol{x}) \neq 0 \to \bot$. *Then* $\models A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{f}(\boldsymbol{x}) \neq 0 \to \bot$.

Proof. By satisfiability there exists $\boldsymbol{y} \in \mathbb{R}^n$ such that $A\boldsymbol{y} \bowtie \boldsymbol{b}$ and $\boldsymbol{a}^T\boldsymbol{y} < b'$.

Let $\boldsymbol{x} \in \mathbb{R}^n$ be arbitrary such that $A\boldsymbol{x} \bowtie \boldsymbol{b}$. We need to show $f_i(\boldsymbol{x}) = 0$ for some i.

For $\lambda \in \mathbb{R}$ let $\boldsymbol{z}_\lambda \coloneqq \lambda \cdot \boldsymbol{x} + (1-\lambda) \cdot \boldsymbol{y}$. Note that for all $0 \leq \lambda < 1$, $A\boldsymbol{z}_\lambda \bowtie \boldsymbol{b}$. Furthermore, there exists $\delta > 0$ satisfying $\boldsymbol{a}^T\boldsymbol{z}_\lambda < b'$ for all $0 \leq \lambda \leq \delta$. Define

$$\begin{aligned} h_i : \mathbb{R} &\to \mathbb{R} \\ \lambda &\mapsto f_i(\boldsymbol{z}_\lambda) \end{aligned}$$

By assumption, for some $1 \leq i \leq k$, h_i needs to have a non-negligible set of roots. Therefore, by Lemma 2.31, $h_i = 0$ for some $1 \leq i \leq k$ and the claim follows since $f_i(\boldsymbol{x}) = h_i(1)$. □

Corollary 8.21. (i) *If* $A\boldsymbol{x} \bowtie \boldsymbol{b}$ *and* $\boldsymbol{a}^T\boldsymbol{x} < b'$ *are satisfiable then the following are equivalent:*

(a) $\models A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{a}^T\boldsymbol{x} < b' \wedge \boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0} \to \bot$
(b) $\models A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0} \to \bot$

(ii) *If* $A\boldsymbol{x} \bowtie \boldsymbol{b}$ *and* $\boldsymbol{a}^T\boldsymbol{x} < b'$ *are satisfiable then the following are equivalent:*

(a) $\models A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{a}^T\boldsymbol{x} \leq b' \wedge \boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0} \to \bot$
(b) $\models A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0} \to \bot$

(iii) *If* $A\boldsymbol{x} \bowtie \boldsymbol{b}$ *is satisfiable, and* $A\boldsymbol{x} \bowtie \boldsymbol{b}$ *and* $\boldsymbol{a}^T\boldsymbol{x} < b'$ *are unsatisfiable (together) then the following are equivalent:*

(a) $\models A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{a}^T\boldsymbol{x} \leq b' \wedge \boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0} \to \bot$
(b) $\models A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{a}^T\boldsymbol{x} = b' \wedge \boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0} \to \bot$

The significance of this is that we can successively remove inequalities or turn them into equalities. Thus, we can reduce deciding $\models A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0} \to \bot$ to deciding $\models A'\boldsymbol{x} = \boldsymbol{b}' \wedge \boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0} \to \bot$, which we have already established.

The randomised decision procedure is summarised in Algorithm 2. Consequently, with Corollary 8.21 and Lemma 8.18 we conclude:

Proposition 8.22. *With probability 1,* $\models A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0} \to \bot$ *iff Algorithm 2 returns true.*

Example 8.23. Consider $z \leq 0 \wedge z \geq 0 \wedge z < 1 \wedge -z^2 + z - 1 \neq z + 1$, which adds the constraint $z < 1$ to the background goal clause derived in Example 8.15. Executing the algorithm, we first remove the redundant constraint $z < 1$. Then the non-strict inequalities are turned into equalities because $z < 0 \wedge z \geq 0$ and $z = 0 \wedge z > 0$ are unsatisfiable, resulting in

$$z = 0 \wedge -z^2 + z - 1 \neq z + 1$$

Finally, $\{z \in \mathbb{R} \mid z = 0\} = \{0\}$ and $-0^2 + 0 - 1 \neq 0 + 1$ and the algorithm correctly rejects the clause.

Data: $\boldsymbol{x} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^k$ are analytic
Result: $\models A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{0} \to \bot$
if $A\boldsymbol{x} \bowtie \boldsymbol{b}$ *is unsatisfiable* **then**
  return true;
end
Remove rows $\boldsymbol{a}_i^T \boldsymbol{x} < b_i$ from $A\boldsymbol{x} \bowtie \boldsymbol{b}$;
for *row* $\boldsymbol{a}^T \boldsymbol{x} \leq b_i$ *in* $A\boldsymbol{x} \bowtie \boldsymbol{b}$ **do**
  /* Invariant: $A\boldsymbol{x} \bowtie \boldsymbol{b}$ is satisfiable */
  if $A\boldsymbol{x} \bowtie' \boldsymbol{b}$ *is satisfiable, where* $\bowtie_i$ *is modified to* $<$ **then**
    remove row;
  else
    replace $\bowtie_i$ with $=$;
  end
end
/* all $\leq$ and $<$ are eliminated: $A\boldsymbol{x} \bowtie \boldsymbol{b}$ is $A\boldsymbol{x} = \boldsymbol{b}$ and it is satisfiable */
Compute rank $r \in \mathbb{N}$, $C \in \mathbb{R}^{n \times (n-r)}$ and $\boldsymbol{d} \in \mathbb{R}^n$ (via e.g. Gaussian elimination) such that $\{\boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} = \boldsymbol{b}\} = \{C\boldsymbol{z} + \boldsymbol{d} \mid \boldsymbol{z} \in \mathbb{R}^{n-r}\}$;
sample $\boldsymbol{z} \sim \boldsymbol{\mathcal{D}}$;
if $f_i(C\boldsymbol{z} + \boldsymbol{d}) = 0$ *for some* i **then**
  return true;
else
  return false;
end

Algorithm 2: Randomised decision procedure for Eq. (8.2)

8.3.2 Recursion-Free HoCHC

Having established the decidability of the background theory, we wish to additionally prove the decidability of satisfiability of a suitable fragment of higher-order constrained Horn clauses. Due to soundness and completeness result (Theorem 8.16) a sufficient condition for decidability of a fragment is that there are only finitely many clauses which can be derived by the proof system.

We consider such a fragment, which is suitable for languages without recursion (such as the one in Chapter 4). For 1st-order Horn clauses a popular assumption is that there are no cyclic dependencies between relational symbols [Rümmer et al., 2015, Unno and Terauchi, 2015, Gupta et al., 2011]. Formally, we define a dependency relation $Q \prec_\Phi P$ between relational symbols Q and P if there exists $B \to P\,\boldsymbol{x}\,t$ in the set of clauses Φ and Q occurs in B. Let $\prec_\Phi^+$ and $\prec_\Phi^*$ be its transitive and reflexive-transitive closure, respectively. In our *higher-order* setting we require:

Definition 8.24. A finite set Φ of HoCHCs is ***recursion-free*** if there are finitely many relational symbols $P_1, \ldots, P_n$ such that each $P_i \prec_\Phi^+ P_j$ implies $i < j$ and for all terms $P_i\,t_1 \cdots t_n$ occurring in Φ, if $P_j \prec_\Phi^* P_i$ and P_k occurs in one of $t_1, \ldots, t_n$ then $k < j$.

Note that the second part vacuously holds for 1st-order clauses and hence recursion-freeness amounts to acyclic dependencies.

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\text{aff}} M_1 : \iota_1^{(\mathbf{f})} \quad \cdots \quad \Gamma \mid \Sigma_\ell \vdash_{\text{aff}} M_\ell : \iota_\ell^{(\mathbf{f})}}{\Gamma \mid \Sigma_1 +\!\!+ \cdots +\!\!+ \Sigma_\ell \vdash_{\text{aff}} \underline{f}(M_1, \dots M_\ell) : \iota^{(\mathbf{f})}} \; f : [\![\iota_1]\!] \times \cdots \times [\![\iota_\ell]\!] \to [\![\iota]\!] \in \mathrm{Op}$$

$$\frac{\Gamma \mid \Sigma_1 \vdash_{\text{aff}} M_1 : R^{(\mathbf{t})} \quad \Gamma \mid \Sigma_2 \vdash_{\text{aff}} M_2 : R^{(\mathbf{t})}}{\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 \vdash_{\text{aff}} \underline{c} \cdot M_1 \underline{+} M_2 : R^{(\mathbf{t})}} \; c \in \mathbb{R}$$

$$\frac{\Gamma \mid \Sigma \vdash_{\text{aff}} L : R^{(\mathbf{t})} \quad \Gamma \mid \Sigma' \vdash_{\text{aff}} M : \sigma \quad \Gamma \mid \Sigma'' \vdash_{\text{aff}} N : \sigma}{\Gamma \mid \Sigma +\!\!+ \Sigma' +\!\!+ \Sigma'' \vdash_{\text{aff}} \mathbf{if}\, L < 0\, \mathbf{then}\, M\, \mathbf{else}\, N : \sigma}$$

$$\frac{}{\boldsymbol{\theta} \mid [\mathcal{D}] \vdash_{\text{aff}} \mathbf{sample}_{\mathcal{D}} \rhd \varphi_{\boldsymbol{\theta}} : R^{(\mathbf{t})}} \; \varphi_{(-)} \in \mathrm{Diff}_{\boldsymbol{\Theta}}(\mathrm{supp}(\mathcal{D}))$$

Figure 8.4: Rules refining the generic type system Fig. 4.5 to guarantee (piecewise) affine guards. (The rules for subtyping, variables, applications and abstractions are unchanged.)

Example 8.25. (i) The set of clauses in Example 8.9 is recursion-free: it only holds Twice, Branch $\prec$ Term. Thus, we can arrange the symbols as Branch, Twice, Term.
(ii) $P\,(2x)\,y \to P\,x\,y$ is not recursion-free (cf. Example 8.41).

To each goal clause G we can identify a vector $\boldsymbol{c}_G := (c_1, \dots, c_n) \in \mathbb{N}^n$, where c_i is the number of occurrences of P_i in G. If $\Phi \vdash_{\text{res}} G'$, where G' is the resolvent of a goal clause G and definite clause in Φ, then $\Phi \cup \{G'\}$ is recursion-free, too, and $\boldsymbol{c}_{G'}$ is lexicographically smaller than $\boldsymbol{c}_G$ because. Consequently, only finitely many goal clauses can be derived.

Finally, applicability of the constraint refutation rule can be decided by Algorithm 2. Consequently:

Theorem 8.26. *Let Φ be a finite set of recursion-free HoCHCs. Φ is unsatisfiable iff $\Phi \vdash_{\text{res}} \bot$, and this is decidable.*

8.4 Encoding

As a warm-up for continuity, we wish to verify that for two terms $\boldsymbol{\theta} \mid \Sigma \vdash_{\text{tr}} M : R$ and $\boldsymbol{\theta} \mid \Sigma \vdash_{\text{tr}} N : R$, $[\![M]\!] = [\![N]\!]$. We achieve this by encoding the input-output-graph of terms (generalising the approach of Example 8.9).

As discussed above, we need to ensure that conditionals only use (piecewise) affine guards to fall into our background theory. We guarantee this by employing another instance of the generic type system of Section 4.6 and by annotating base types $R^{(a)}$ and $R^{(a)}_{>0}$, where $a \in \{\mathbf{t}, \mathbf{f}\}$ denotes whether the term is (piecewise) affine. The subtyping relation is $\iota_1^{(a_1)} \sqsubseteq \iota_2^{(a_2)}$ if $\iota_1 \sqsubseteq \iota_2$ and $a_1 = \mathbf{f}$ implies $a_2 = \mathbf{f}$, and extended to higher order as in Eq. (4.2).

We refine the generic type system of Section 4.6 in Fig. 8.4.

To generalise the encoding in Example 8.9 we stipulate:

$$\rho_\iota := \iota \to o$$

$$\rho_{\iota \bullet \Sigma \to \tau} \coloneqq \iota \to \underbrace{R^{(\mathbf{t})} \to \cdots \to R^{(\mathbf{t})}}_{|\Sigma| \text{ times}} \to \rho_\tau$$

$$\rho_{\tau_1 \bullet \Sigma \to \tau_2} \coloneqq \rho_{\tau_1} \to \underbrace{R^{(\mathbf{t})} \to \cdots \to R^{(\mathbf{t})}}_{|\Sigma| \text{ times}} \to \rho_{\tau_2} \qquad (\tau_1 \neq \iota)$$

$$\sigma_\iota \coloneqq \iota$$

$$\sigma_\tau \coloneqq \rho_\tau \qquad (\tau \neq \iota)$$

For a term-in-context, $x_1 : \tau_1, \ldots, x_m : \tau_m \mid [\mathcal{D}_1, \ldots, \mathcal{D}_n] \vdash M : \tau$ we introduce relational symbols

$$\mathrm{Gr}_{x_1:\tau_1,\ldots,x_m:\tau_m \mid [\mathcal{D}_1,\ldots,\mathcal{D}_n] \vdash M:\tau} : \sigma_{\tau_1} \to \cdots \to \sigma_{\tau_m} \to \underbrace{R^{(\mathbf{t})} \to \cdots \to R^{(\mathbf{t})}}_{n \text{ times}} \to \rho_\tau$$

For example for a term $x : R^{(\mathbf{t})} \mid [\mathcal{N}] \vdash_{\text{aff}} M : R^{(\mathbf{t})}$,

$$[\![M]\!] : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \qquad \mathrm{Gr}_{x:R^{(\mathbf{t})} \mid [\mathcal{N}] \vdash_{\text{aff}} M:R^{(\mathbf{t})}} : R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o$$

To enhance readability we henceforth suppress the context and type information and just write Gr_M.

Then, we axiomatise the graph of $[\![M]\!]^t$ (cf. Section 4.2.1) by adding the corresponding axiomatisations in Fig. 8.5 of all subterms of M to the set Φ_M of definite clauses. In particular for constants $r \in \mathbb{R}$, $\to \mathrm{Gr}_{\underline{r}}\, \boldsymbol{x}\, r$. In the rules marked by $\star$ we omit variables needed on both sides to obtain a foreground atom, which has type o. For instance, if $\Gamma \mid [] \vdash_{\text{aff}} M : R^{(\mathbf{t})} \bullet [\mathcal{N}] \to R^{(\mathbf{t})}$ and $\Gamma \mid [] \vdash_{\text{aff}} N : R^{(\mathbf{t})}$ then

$$\mathrm{Gr}_N\, \boldsymbol{x}\, y \wedge \mathrm{Gr}_M\, \boldsymbol{x}\, y\, z \to \mathrm{Gr}_{M\,N}\, \boldsymbol{x}\, z$$

is an abbreviation for

$$\mathrm{Gr}_N\, \boldsymbol{x}\, y \wedge \mathrm{Gr}_M\, \boldsymbol{x}\, z\, y\ v \to \mathrm{Gr}_{M\,N}\, \boldsymbol{x}\, z\ v$$

where $v : R^{(\mathbf{t})} \in \Delta$.

Example 8.27. We list the typing judgements, introduced relational symbols and axiomatisations of all subterms of the following term in Table 8.1:

$$\underbrace{(\lambda f.\, f\,(f\,0))}_{\equiv M}\ \underbrace{(\lambda x.\, x + \mathbf{sample}_{\mathcal{N}} \rhd \mathrm{id})}_{\equiv N}$$

By inspecting the rules, we can easily see that Φ_M is a set of recursion-free definite clauses because relational symbols (corresponding to subterms of M) can be numbered according to the reverse of the pre-order traversal of M's syntax tree.

Now, in order to prove equivalence of $\Gamma \mid \Sigma \vdash_{\text{aff}} M : R$ and $\Gamma \mid \Sigma \vdash_{\text{aff}} N : R$ we add the goal clause

$$\mathrm{Gr}_M\, \boldsymbol{x}\, \boldsymbol{z}\, y \wedge \mathrm{Gr}_N\, \boldsymbol{x}\, \boldsymbol{z}\, y' \wedge y \neq y' \to \bot$$

to $\Phi_M \cup \Phi_N$, and call the resulting clause set $\Phi^{=}_{M,N}$. We would like to show the following:

Table 8.1: Typing judgements, introduced relational symbols and axiomatisations of all subterms for Example 8.27. The variables have sorts $u, u_1, u_2, u_3, y, y', z, z_1, z_2 : R^{(\mathbf{t})}$ and $p : R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o$. For the sake of completeness we have added the variables, which were omitted in Fig. 8.5 and wavy-underline them.

$f : R^{(\mathbf{t})} \bullet [\mathcal{N}] \to R^{(\mathbf{t})} \mid [] \vdash_{\text{aff}} 0 : R^{(\mathbf{t})}$
$\rightsquigarrow \mathrm{Gr}_0 : (R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o) \to R^{(\mathbf{t})} \to o$
$\to \mathrm{Gr}_0\, p\, 0$

$f : R^{(\mathbf{t})} \bullet [\mathcal{N}] \to R^{(\mathbf{t})} \mid [] \vdash_{\text{aff}} f : R^{(\mathbf{t})} \to R^{(\mathbf{t})}$
$\rightsquigarrow \mathrm{Gr}_f : (R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o) \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o$
$p\, u_1\, u_2\, u_3 \to \mathrm{Gr}_f\, p\, u_1\, u_2\, u_3$

$f : R^{(\mathbf{t})} \bullet [\mathcal{N}] \to R^{(\mathbf{t})} \mid [\mathcal{N}] \vdash_{\text{aff}} f\, 0 : R^{(\mathbf{t})}$
$\rightsquigarrow \mathrm{Gr}_{f\,0} : (R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o) \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o$
$\mathrm{Gr}_0\, p\, y \wedge \mathrm{Gr}_f\, p\, y\, z_1\, u \to \mathrm{Gr}_{f\,0}\, p\, z_1\, u$

$f : R^{(\mathbf{t})} \bullet [\mathcal{N}] \to R^{(\mathbf{t})} \mid [\mathcal{N}, \mathcal{N}] \vdash_{\text{aff}} f\,(f\,0) : R^{(\mathbf{t})}$
$\rightsquigarrow \mathrm{Gr}_{f\,(f\,0)} : (R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o) \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o$
$\mathrm{Gr}_{f\,0}\, p\, z_1\, y \wedge \mathrm{Gr}_f\, p\, y\, z_2\, u \to \mathrm{Gr}_{f\,(f\,0)}\, p\, z_1\, z_2\, u$

$\mid [] \vdash_{\text{aff}} \lambda f.\, f\,(f\,0) : (R^{(\mathbf{t})} \to R^{(\mathbf{t})}) \bullet [\mathcal{N}, \mathcal{N}] \to R^{(\mathbf{t})}$
$\rightsquigarrow \mathrm{Gr}_{\lambda f.f\,(f\,0)} : (R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o) \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o$
$\mathrm{Gr}_{f\,(f\,0)}\, p\, u_1\, u_2\, u_3 \to \mathrm{Gr}_{\lambda f.f\,(f\,0)}\, p\, u_1\, u_2\, u_3$

$x : R^{(\mathbf{t})} \mid [\mathcal{N}] \vdash_{\text{aff}} \mathbf{sample}_{\mathcal{N}} \triangleright \mathrm{id} : R^{(\mathbf{t})}$
$\rightsquigarrow \mathrm{Gr}_{\mathbf{sample}_{\mathcal{N}} \triangleright \mathrm{id}} : R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o$
$\to \mathrm{Gr}_{\mathbf{sample}_{\mathcal{N}} \triangleright \mathrm{id}}\, x\, z\, z$

$x : R^{(\mathbf{t})} \mid [] \vdash_{\text{aff}} x : R^{(\mathbf{t})}$
$\rightsquigarrow \mathrm{Gr}_x : R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o$
$\to \mathrm{Gr}_x\, x\, x$

$x : R^{(\mathbf{t})} \mid [\mathcal{N}] \vdash_{\text{aff}} x + \mathbf{sample}_{\mathcal{N}} \triangleright \mathrm{id} : R^{(\mathbf{t})}$
$\rightsquigarrow \mathrm{Gr}_{x + \mathbf{sample}_{\mathcal{N}} \triangleright \mathrm{id}} : R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o$
$\mathrm{Gr}_x\, x\, y \wedge \mathrm{Gr}_{\mathbf{sample}_{\mathcal{N}} \triangleright \mathrm{id}}\, x\, z\, y' \to \mathrm{Gr}_{x + \mathbf{sample}_{\mathcal{N}} \triangleright \mathrm{id}}\, x\, z\, (y + y')$

$\mid [] \vdash_{\text{aff}} \lambda x.\, x + \mathbf{sample}_{\mathcal{N}} \triangleright \mathrm{id} : R^{(\mathbf{t})} \bullet [\mathcal{N}] \to R^{(\mathbf{t})}$
$\rightsquigarrow \mathrm{Gr}_{\lambda x.x + \mathbf{sample}_{\mathcal{N}} \triangleright \mathrm{id}} : R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o$
$\mathrm{Gr}_{x + \mathbf{sample}_{\mathcal{N}} \triangleright \mathrm{id}}\, u_1\, u_2\, u_3 \to \mathrm{Gr}_N\, u_1\, u_2\, u_3$

$\mid [\mathcal{N}, \mathcal{N}] \vdash_{\text{aff}} M\, N : R^{(\mathbf{t})}$
$\rightsquigarrow \mathrm{Gr}_{M\,N} : R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to R^{(\mathbf{t})} \to o$
$\mathrm{Gr}_M\, \mathrm{Gr}_N\, z_1\, z_2\, u \to \mathrm{Gr}_{M\,N}\, z_1\, z_2\, u$

$$
\begin{array}{lll}
x_1 : \tau_1, \dots, x_n : \tau_n \mid [] \vdash_{\text{aff}} x_i : \tau_i & & \\
\qquad \to \mathrm{Gr}_{x_i}\, \boldsymbol{x}\, x_i & & \text{if } \tau_i = \iota \\
\star \qquad x_i \to \mathrm{Gr}_{x_i}\, \boldsymbol{x} & & \text{otherwise} \\
\Gamma \mid \Sigma_1 + \!\!+ \cdots + \!\!+ \Sigma_\ell \vdash_{\text{aff}} \underline{f}(M_1, \dots, M_\ell) : \iota & & \\
\qquad \mathrm{Gr}_{M_1}\, \boldsymbol{x}\, \boldsymbol{z}_1\, y_1 \wedge \ldots \wedge \mathrm{Gr}_{M_\ell}\, \boldsymbol{x}\, \boldsymbol{z}_\ell\, y_\ell \to \mathrm{Gr}_{\underline{f}(M_1,\dots,M_\ell)}\, \boldsymbol{x}\, \boldsymbol{z}_1 \cdots \boldsymbol{z}_\ell\, (f(y_1, \dots, y_\ell)) & & \\
\Gamma \mid \Sigma_1 + \!\!+ \Sigma_2 + \!\!+ \Sigma_3 \vdash_{\text{aff}} \mathbf{if}\, L < 0\, \mathbf{then}\, M\, \mathbf{else}\, N : \iota & & \\
\qquad \mathrm{Gr}_L\, \boldsymbol{x}\, \boldsymbol{z}_1\, y' \wedge y' < 0 \wedge \mathrm{Gr}_M\, \boldsymbol{x}\, \boldsymbol{z}_2\, y \to \mathrm{Gr}_{\mathbf{if}\, L<0\, \mathbf{then}\, M\, \mathbf{else}\, N}\, \boldsymbol{x}\, \boldsymbol{z}_1\, \boldsymbol{z}_2\, \boldsymbol{z}_3\, y & & \\
\qquad \mathrm{Gr}_L\, \boldsymbol{x}\, \boldsymbol{z}_1\, y' \wedge y' \geq 0 \wedge \mathrm{Gr}_N\, \boldsymbol{x}\, \boldsymbol{z}_3\, y \to \mathrm{Gr}_{\mathbf{if}\, L<0\, \mathbf{then}\, M\, \mathbf{else}\, N}\, \boldsymbol{x}\, \boldsymbol{z}_1\, \boldsymbol{z}_2\, \boldsymbol{z}_3\, y & & \\
\Gamma \mid [] \vdash_{\text{aff}} \lambda y.\, M : \tau_1 \bullet \Sigma \to \tau_2 & & \\
\star \qquad \mathrm{Gr}_M \to \mathrm{Gr}_{\lambda y.M} & & \\
\Gamma \mid \Sigma_1 + \!\!+ \Sigma_2 + \!\!+ \Sigma_3 \vdash_{\text{aff}} M\, N : \tau \qquad (\text{and } \Gamma \mid \Sigma_1 \vdash_{\text{aff}} M : \tau' \bullet \Sigma_3 \to \tau) & & \\
\star \qquad \mathrm{Gr}_N\, \boldsymbol{x}\, \boldsymbol{z}_2\, y \wedge \mathrm{Gr}_M\, \boldsymbol{x}\, \boldsymbol{z}_1\, y\, \boldsymbol{z}_3 \to \mathrm{Gr}_{M\,N}\, \boldsymbol{x}\, \boldsymbol{z}_1\, \boldsymbol{z}_2\, \boldsymbol{z}_3 & & \text{if } \tau' = \iota \\
\star \qquad \mathrm{Gr}_M\, \boldsymbol{x}\, \boldsymbol{z}_1\, (\mathrm{Gr}_N\, \boldsymbol{x}\, \boldsymbol{z}_2)\, \boldsymbol{z}_3 \to \mathrm{Gr}_{M\,N}\, \boldsymbol{x}\, \boldsymbol{z}_1\, \boldsymbol{z}_2\, \boldsymbol{z}_3 & & \text{otherwise} \\
\boldsymbol{\theta}, \Gamma \mid [\mathcal{D}] \vdash_{\text{aff}} \mathbf{sample}_{\mathcal{D}} \rhd \boldsymbol{\varphi}_{\boldsymbol{\theta}} : R^{(\mathbf{t})} & & \\
\qquad \to \mathrm{Gr}_{\mathbf{sample}_{\mathcal{D}} \rhd \boldsymbol{\varphi}_{\boldsymbol{\theta}}}\, \boldsymbol{x}\, z\, z & &
\end{array}
$$

Figure 8.5: Axiomatisation of the graph

Proposition 8.28. $[\![M]\!] = [\![N]\!]$ *iff* $\Phi^{=}_{M,N}$ *is satisfiable.*

As a consequence, $[\![M]\!] = [\![N]\!]$ is decidable because $\Phi^{=}_{M,N}$ is recursion-free and hence its satisfiability is decidable.

To demonstrate the if-direction, we prove that every model of Φ_M and Φ_N over-approximates their graph:

Lemma 8.29. *Suppose* $\boldsymbol{\theta} \mid \Sigma \vdash_{\text{aff}} M : \iota^{(a)}$ *and* $\mathcal{A} \models \Phi_M$. *If* $[\![M]\!](\boldsymbol{\theta}, \boldsymbol{z}) = r$ *then* $R^{\mathcal{A}}(\boldsymbol{\theta})(\boldsymbol{z})(r) = 1$.

To prove the claim, we posit a logical relation $\preceq_\tau$ on $[\![\tau]\!] \times [\![\rho_\tau]\!]$:

(i) $r \preceq_\iota p$ if $p(r) = 1$

(ii) $f \preceq_{\iota \bullet \Sigma \to \tau} p$ if for all $r \in [\![\iota]\!]$ and $z_1, \dots, z_{|\Sigma|} \in \mathbb{R}$,

$$f(r, [z_1, \dots, z_{|\Sigma|}]) \preceq_\tau p(r)(z_1) \cdots (z_{|\Sigma|})$$

(iii) $f \preceq_{\tau_1 \bullet \Sigma \to \tau_2} p$, where $\tau_1 \neq \iota$, if for all $g \preceq_{\tau_1} q$ and $z_1, \dots, z_{|\Sigma|} \in \mathbb{R}$,

$$f\left(g, [z_1, \dots, z_{|\Sigma|}]\right) \preceq_{\tau_2} p(q)(z_1) \cdots (z_{|\Sigma|})$$

Besides, we prove a fundamental lemma (see Appendix D.3 for a proof):

Lemma 8.30 (Fundamental)**.** *Suppose* $x_1 : \tau_1, \dots, x_m : \tau_m \mid [\mathcal{D}_1, \dots, \mathcal{D}_n] \vdash_{\text{aff}} M : \tau$ *and* $\mathcal{A} \models \Phi_M$. *Then for all* $f_1 \preceq'_{\tau_1} p_1, \dots, f_n \preceq'_{\tau_n} p_n$ *and* $z_1, \dots, z_n \in \mathbb{R}$,

$$[\![M]\!]^t(f_1, \dots, f_m, z_1, \dots, z_n) \preceq_\tau R^{\mathcal{A}}_M(p_1) \cdots (p_m)(z_1) \cdots (z_n)$$

where $r \preceq'_\iota r'$ *iff* $r = r'$ *and for* $\tau \neq \iota$, $f \preceq'_\tau p$ *iff* $f \preceq_\tau p$.

Conversely, to establish the only-if direction of Proposition 8.28, we provide a structure $\mathcal{A} \models \Phi_M$, satisfying the following: $\mathrm{Gr}_M^{\mathcal{A}}(\boldsymbol{\theta})(\boldsymbol{z})(r) = 1$ implies $[\![M]\!]^t(\boldsymbol{\theta}, \boldsymbol{z}) = r$. Intuitively, this forces us to be conservative: heads $P\, x_1 \cdots x_n\, M$ of definite clauses are satisfied precisely if they need to be true because all atoms in the body of a definite clause are satisfied.

Formally, we iteratively define the ***canonical structure*** $\mathcal{A}_\Phi$ (see [Ong and Wagner, 2019]) associated to a recursion-free set Φ of clauses (recall that $A_1, \ldots, A_m$ in definite clauses $A_1 \wedge \cdots \wedge A_m \to P_i\, x_1 \cdots x_n\, M$ only use relational symbols amongst $P_1, \ldots, P_{i-1}$):

$$P_i^{\mathcal{A}_\Phi}(p_1, \ldots, p_m, r) \coloneqq \begin{cases} 1 & \text{if for some } A_1 \wedge \cdots \wedge A_m \to P_i\, x_1 \cdots x_n\, M \in \Phi \text{ and } \alpha, \\ & \qquad \mathcal{A}_\Phi, \alpha \models A_1, \ldots, \mathcal{A}_\Phi, \alpha \models A_k \text{ and} \\ & \qquad \alpha(x_1) = p_1, \ldots, \alpha(x_m) = p_m \text{ and } \mathcal{A}_\Phi[\![M]\!](\alpha) = r \\ 0 & \text{otherwise} \end{cases}$$

Note that by definition $\mathcal{A}_\Phi$ satisfies all definite clauses $D \in \Phi$ [Ong and Wagner, 2019].

The canonical structure satisfies:

Lemma 8.31. *Let* $\boldsymbol{\theta} \mid \Sigma \vdash_{\mathrm{aff}} M : \iota^{(a)}$. *If* $P_M^{\mathcal{A}_{\Phi_M}}(\boldsymbol{\theta})(\boldsymbol{z})(r) = 1$ *then* $[\![M]\!]^t(\boldsymbol{\theta}, \boldsymbol{z}) = r$.

This result can be shown analogously to Lemma 8.29 by positing a modified logical relation $\succeq_\tau$, whereby $r \succeq_\iota p$ iff $p(r') = 1$ implies $r' = r$ and proving a fundamental lemma, which is a minor and straightforward variation of Lemma 8.30.

8.4.1 Overapproximating Continuity

Suppose $\boldsymbol{\theta} \mid \Sigma \vdash_{\mathrm{aff}} M : R$. Recall that by Theorem 8.5 the reparameterisation gradient estimator is unbiased if $[\![M]\!] : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$ is continuous (and $\boldsymbol{\theta} \mid \Sigma \vdash_{\mathrm{int}} M : R$). Therefore, we now turn to encoding continuity in our logic.

Example 8.32. Consider the term:

$$M \equiv \mathbf{if}\, z < 0\, \mathbf{then}\, -z^2 + z - 1\, \mathbf{else}\, z + 1$$

$[\![M]\!] : \mathbb{R} \to \mathbb{R}$ is discontinuous because

$$\lim_{z \nearrow 0} -z^2 + z - 1 = -1 \neq 1 = \lim_{z \searrow 0} z + 1$$

and the two branches do not agree on the boundary $z = 0$.

Instead of axiomatising the graph of M as in Fig. 8.5, we can axiomatise its *closure* (in the relational symbol $\mathrm{GC} : R^{(\mathbf{t})} \to R^{(\mathbf{f})} \to o$):

$$\begin{aligned} z \leq 0 &\to \mathrm{GC}\, z\, (-z^2 + z - 1) \\ z \geq 0 &\to \mathrm{GC}\, z\, (z + 1) \end{aligned}$$

and require $\mathrm{GC}\, z\, y \wedge \mathrm{GC}\, z\, y' \wedge y \neq y' \to \bot$. (Recall that we have refuted these clauses in Example 8.15.)

In general, let Φ_M^{c} be the clause set similar to above where the axiomatisation of conditionals is replaced with (and we use the symbols GC instead of Gr)

$$\mathrm{GC}_L\, \boldsymbol{x}\, \boldsymbol{z}_1\, y' \wedge y' \leq 0 \wedge \mathrm{GC}_M\, \boldsymbol{x}\, \boldsymbol{z}_2\, y \to \mathrm{GC}_{\mathbf{if}\, L < 0\, \mathbf{then}\, M\, \mathbf{else}\, N}\, \boldsymbol{x}\, \boldsymbol{z}_1\, \boldsymbol{z}_2\, \boldsymbol{z}_3\, y \tag{8.6}$$

$$\mathrm{GC}_L\,\boldsymbol{x}\,\boldsymbol{z}_1\,y' \wedge y' \geq 0 \wedge \mathrm{GC}_N\,\boldsymbol{x}\,\boldsymbol{z}_3\,y \to \mathrm{GC}_{\mathbf{if}\,L<0\,\mathbf{then}\,M\,\mathbf{else}\,N}\,\boldsymbol{x}\,\boldsymbol{z}_1\,\boldsymbol{z}_2\,\boldsymbol{z}_3\,y$$

together with the goal clause

$$\mathrm{GC}_M\,\boldsymbol{\theta}\,\boldsymbol{z}\,y \wedge \mathrm{GC}_M\,\boldsymbol{\theta}\,\boldsymbol{z}\,y' \wedge y \neq y' \to \bot$$

Proposition 8.33. *If Φ^{c}_M is satisfiable then $[\![M]\!]$ is continuous.*

Crucially, as a consequence of Theorem 8.26, satisfiability of Φ^{c}_M is decidable, and we thus have developed a method for verifying continuity. (Proposition 8.33 will be corollary of Theorem 8.40.)

8.5 Reasoning about Limits

Adding non-strict in-equalities for both branches in the encoding of conditionals

$$\mathbf{if}\,L < 0\,\mathbf{then}\,M\,\mathbf{else}\,N$$

is clearly an overapproximation. The following example illustrates that it may prevent us from recognising continuous functions as such:

Example 8.34. Consider

$$M \equiv \mathbf{if}\,z < 0\,\mathbf{then}\,0\,\mathbf{else}\,(\mathbf{if}\,z < 0\,\mathbf{then}\,-1\,\mathbf{else}\,z)$$

Observe, that the branch with -1 is actually not reachable and $[\![M]\!](z) = \mathrm{ReLU}(z)$, which is continuous. The closure of its graph and continuity can be axiomatised as in Section 8.4.1 as follows:

$$\begin{aligned} x \leq 0 &\to \mathrm{GC}\,x\,0 \\ x \geq 0 \wedge x \leq 0 &\to \mathrm{GC}\,x\,(-1) \\ x \geq 0 \wedge x \geq 0 &\to \mathrm{GC}\,x\,x \\ \mathrm{GC}\,z\,y \wedge \mathrm{GC}\,z\,y' \wedge y \neq y' &\to \bot \end{aligned}$$

Performing two resolution inferences of the goal clause with the first and second clause, we can derive:

$$z \leq 0 \wedge z \geq 0 \wedge z \leq 0 \wedge 0 \neq -1 \to \bot$$

which is not valid (take the valuation $z \mapsto 0$), and hence can be refuted.

In contrast to Eq. (8.6), we can refrain from axiomatising the closure of the graph:

$$\begin{aligned} x < 0 &\to \mathrm{Gr}\,x\,0 \\ x \geq 0 \wedge x < 0 &\to \mathrm{Gr}\,x\,(-1) \\ x \geq 0 \wedge x \geq 0 &\to \mathrm{Gr}\,x\,x \end{aligned}$$

Continuity then boils down to requiring that for all *sequences* $(z_k)_{k\in\mathbb{N}}$ and $(y_k)_{k\in\mathbb{N}}$:

$$\mathrm{Gr}\,z\,y \wedge \lim_{k\to\infty} z_k = z \wedge (\forall k \in \mathbb{N}.\,\mathrm{Gr}\,z_k\,y_k) \to y = \lim_{k\to\infty} y_k \tag{8.7}$$

In the next subsection we propose a novel semantics based on *sequences* of valuations and introduce new predicates $\Downarrow$ and $\not\Downarrow$ to reason about (common) limits. In particular, we modify (8.7) to

$$\mathrm{Gr}\,z\,y \wedge z \Downarrow z' \wedge \mathrm{Gr}\,z'\,y' \wedge y \not\Downarrow y' \to \bot \tag{8.8}$$

8.5.1 Background Theory and Semantics

In general, we introduce novel predicates $\Downarrow$ and $\not\Downarrow$:

$$\frac{}{\Delta \vdash x \Downarrow y : o}\; x : \iota_1^{(\mathbf{t})}, y : \iota_2^{(\mathbf{t})} \in \Delta \qquad \frac{\Delta \vdash t_1 : R^{(\mathbf{f})} \quad \Delta \vdash t_2 : R^{(\mathbf{f})}}{\Delta \vdash t_1 \not\Downarrow t_2 : o}$$

along with

$$\frac{t_1 : R^{(\mathbf{t})} \quad t_2 : R^{(\mathbf{t})}}{t_1 \bowtie t_2 : o}\; \bowtie \in \{=, \leq, <\}$$

To simplify the presentation we henceforth disallow disequations of analytic, non-affine terms. (This is not a significant restriction because our revised encoding uses $\not\Downarrow$ instead of $\neq$; see Eq. (8.8).)

We slightly modify the definition of clauses[7]:

- A definite clause neither contains $\Downarrow$ nor $\not\Downarrow$.
- In a goal clause $x \Downarrow x' \wedge \varphi_1 \wedge \cdots \wedge \varphi_\ell \wedge t_1\, y_1 \wedge \cdots \wedge t_m\, y_m \to \bot$, x and x' are distinct from $y_1, \ldots, y_m$.

Thus, we can leave the satisfaction relation $\models_{\lim}$ for definite clauses D unchanged: $\mathcal{A} \models_{\lim} D$ if $\mathcal{A} \models D$. For goal clauses $\models_{\lim}$ is based on *convergent sequences* $(\alpha_k)_{k\in\mathbb{N}}$ of valuations, i.e. for $x : \iota$, $\lim \alpha_k(x)$ exists. (Recall that goal clauses only have variables of base sort.) We stipulate:

- $(\alpha_k)_{k\in\mathbb{N}} \models_{\lim} x \Downarrow y$ iff $\lim_{k\to\infty} \alpha_k(x) = \lim_{k\to\infty} \alpha_k(y)$
- $(\alpha_k)_{k\in\mathbb{N}} \models_{\lim} t_1 \not\Downarrow t_2$ iff $\lim_{k\to\infty} [\![t_1]\!](\alpha_k) \neq \lim_{k\to\infty} [\![t_2]\!](\alpha_k)$

NB Since all primitives are continuous, $[\![t]\!]$ is continuous for $\Delta \vdash t : \iota^{(a)}$.

Besides, for a foreground atom A, $\mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} A$ iff for all $k \in \mathbb{N}$, $\mathcal{A}, \alpha_k \models_{\lim} A$. Finally, for a goal clause $\mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} A_1 \wedge \cdots \wedge A_\ell \to \bot$ if for some $1 \leq i \leq \ell$, $\mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \not\models_{\lim} A_i$, and $\mathcal{A} \models_{\lim} G$ if for all convergent $(\alpha_k)_{k\in\mathbb{N}}$, $\mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} G$. For a set of clauses Φ, $\mathcal{A} \models_{\lim} \Phi$ if for all $G \in \Phi$, $\mathcal{A} \models_{\lim} G$ and for all $D \in \Phi$, $\mathcal{A} \models D$. In this case we call Φ ***lim-satisfiable***.

Example 8.35. (i) $\not\models_{\lim} x < 0 \wedge x' \geq 0 \wedge x \Downarrow x' \wedge 0 \not\Downarrow 1 \to \bot$
(ii) $\models_{\lim} x < 0 \wedge x' \geq 0 \wedge x \Downarrow x' \wedge -x \not\Downarrow x' \to \bot$
(iii) $\models_{\lim} z < 0 \wedge z \geq 0 \wedge z' < 0 \wedge z \Downarrow z' \wedge 0 \not\Downarrow -1 \to \bot$

The last background clause is the only one we can derive in the modified encoding of Example 8.34 exhibiting the potential for refutation due to the conjunct $0 \not\Downarrow -1$. Thus, using the modified encoding, the continuous term of Example 8.34 is not unnecessarily rejected.

[7]The additional restriction will prove important to obtain a decidable background theory. In particular the restriction on goal clauses rules out $y \Downarrow z \wedge P\, y\, z \to \bot$, which together with $\to P\, x\, (f(x))$ might resolve to $y \Downarrow f(y) \to \bot$. This is not covered by Proposition 8.38, which is the key to decidability.

8.5.2 Encoding

To encode continuity of a term $\boldsymbol{\theta} \mid \Sigma \vdash_{\text{aff}} M : R$ we add the goal clause[8]

$$\mathrm{Gr}_M\, \boldsymbol{\theta}\, \boldsymbol{z}\, y \wedge \mathrm{Gr}_M\, \boldsymbol{\theta}'\, \boldsymbol{z}'\, y' \wedge \boldsymbol{\theta} \Downarrow \boldsymbol{\theta}' \wedge \boldsymbol{z} \Downarrow \boldsymbol{z}' \wedge y \not\Downarrow y' \to \bot$$

to Φ_M, and we call the resulting clause set $\Phi_M^{\lim}$.

Example 8.36. Now consider, the (discontinuous)

$$\mathbf{if}\, z < 0\, \mathbf{then}\, -1\, \mathbf{else}\, z$$

This results in the following clause

$$\begin{aligned} z < 0 &\to \mathrm{Gr}\, z\, (-1) \\ z \geq 0 &\to \mathrm{Gr}\, z\, z \\ \mathrm{Gr}\, z\, y \wedge \mathrm{Gr}\, z'\, y' \wedge z \Downarrow z' \wedge y \not\Downarrow y' &\to \bot \end{aligned}$$

Note that for any structure $\mathcal{A}$ satisfying the first two clauses, $\mathrm{Gr}^{\mathcal{A}}(z)(y) = 1$ if $z < 0$ and $y = -1$, or $z = y \geq 0$. Then for $(\alpha_k)_{k\in\mathbb{N}}$ defined by

$$\alpha_k(z) \coloneqq -1/k \qquad \alpha_k(y) \coloneqq -1 \qquad \alpha_k(z') \coloneqq 1/k \qquad \alpha_k(y') \coloneqq 1/k$$

the goal clause is not satisfied because

$$\begin{aligned} &\mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} \mathrm{Gr}\, z\, y && \mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} \mathrm{Gr}\, z'\, y' \\ &\mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} z \Downarrow z' && \mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} y \not\Downarrow y' \end{aligned}$$

Proposition 8.37 (Correctness of Encoding)**.** *$[\![M]\!]$ is continuous iff $\Phi_M^{\lim}$ is lim-satisfiable.*

Proof. By Proposition 5.6, it is sufficient to show the claim for $[\![M]\!]^t : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$.

Suppose $\Phi_M^{\lim}$ is lim-satisfiable. Then there exists $\mathcal{A} \models_{\lim} \Phi_M^{\lim}$. To show continuity of $[\![M]\!]^t$, suppose $\boldsymbol{r}_k \to \boldsymbol{r}$ and $\boldsymbol{s}_k \to \boldsymbol{s}$. By Lemma 8.29, $\mathrm{Gr}_M^{\mathcal{A}}(\boldsymbol{r}_k)(\boldsymbol{s}_k)([\![M]\!]^t(\boldsymbol{r}_k, \boldsymbol{s}_k)) = 1$ for all $k \in \mathbb{N}$ and $\mathrm{Gr}_M^{\mathcal{A}}(\boldsymbol{r})(\boldsymbol{s})([\![M]\!]^t(\boldsymbol{r}, \boldsymbol{s})) = 1$. Let α_k be defined (pointwise) by

$$\begin{aligned} \alpha_k(\boldsymbol{\theta}) &\coloneqq \boldsymbol{r}_k & \alpha_k(\boldsymbol{z}) &\coloneqq \boldsymbol{s}_k & \alpha_k(y) &\coloneqq [\![M]\!]^t(\boldsymbol{r}_k, \boldsymbol{s}_k) \\ \alpha_k(\boldsymbol{\theta}') &\coloneqq \boldsymbol{r} & \alpha_k(\boldsymbol{z}') &\coloneqq \boldsymbol{s} & \alpha_k(y') &\coloneqq [\![M]\!]^t(\boldsymbol{r}, \boldsymbol{s}) \end{aligned}$$

Clearly,

$$\begin{aligned} &\mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} \boldsymbol{\theta} \Downarrow \boldsymbol{\theta}' && \mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} \boldsymbol{z} \Downarrow \boldsymbol{z}' \\ &\mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} \mathrm{Gr}_M\, \boldsymbol{\theta}\, \boldsymbol{z}\, y && \mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} \mathrm{Gr}_M\, \boldsymbol{\theta}'\, \boldsymbol{z}'\, y' \end{aligned}$$

Therefore, by assumption, $\mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \not\models_{\lim} y \not\Downarrow y'$, i.e.

$$\lim [\![M]\!]^t(\boldsymbol{r}_k, \boldsymbol{s}_k) = \lim \alpha_k(y) = \lim \alpha_k(y') = [\![M]\!]^t(\boldsymbol{r}, \boldsymbol{s}) = [\![M]\!]^t(\lim \boldsymbol{r}_k, \lim \boldsymbol{s}_k)$$

and $[\![M]\!]^t$ is continuous.

[8] instead of $\mathrm{GC}_M\, \boldsymbol{\theta}\, \boldsymbol{z}\, y \wedge \mathrm{GC}_M\, \boldsymbol{\theta}\, \boldsymbol{z}\, y' \wedge y \neq y' \to \bot$

Conversely, suppose $[\![M]\!]^t$ is continuous. Recall that for the canonical structure $\mathcal{A}_M \models \Phi_M$. Therefore, it suffices to show,

$$\mathcal{A}_M \models_{\lim} \mathrm{Gr}_M\, \boldsymbol{\theta}\, \boldsymbol{z}\, y \wedge \mathrm{Gr}_M\, \boldsymbol{\theta}'\, \boldsymbol{z}'\, y' \wedge \boldsymbol{\theta} \Downarrow \boldsymbol{\theta}' \wedge \boldsymbol{z} \Downarrow \boldsymbol{z}' \wedge y \not\Downarrow y' \to \bot$$

Let $(\alpha_k)_{k\in\mathbb{N}}$ be an arbitrary convergent sequence of valuations satisfying

$$\mathcal{A}_M, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} \mathrm{Gr}_M\, \boldsymbol{\theta}\, \boldsymbol{z}\, y \qquad \mathcal{A}_M, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} \mathrm{Gr}_M\, \boldsymbol{\theta}'\, \boldsymbol{z}'\, y'$$
$$\mathcal{A}_M, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} \boldsymbol{\theta} \Downarrow \boldsymbol{\theta}' \qquad \mathcal{A}_M, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} \boldsymbol{z} \Downarrow \boldsymbol{z}'$$

By definition, $\lim_k \alpha_k(\boldsymbol{\theta}) = \lim_k \alpha_k(\boldsymbol{\theta}')$ and $\lim_k \alpha_k(\boldsymbol{z}) = \lim_k \alpha_k(\boldsymbol{z}')$ (component-wise) and by Lemma 8.31, for every $k \in \mathbb{N}$,

$$[\![M]\!]^t(\alpha_k(\boldsymbol{\theta}), \alpha_k(\boldsymbol{z})) = \alpha_k(y) \qquad [\![M]\!]^t(\alpha_k(\boldsymbol{\theta}'), \alpha_k(\boldsymbol{z}')) = \alpha_k(y')$$

Therefore, by continuity of $[\![M]\!]^t$,

$$\begin{aligned}\lim \alpha_k(y) &= \lim [\![M]\!]^t(\alpha_k(\boldsymbol{\theta}), \alpha_k(\boldsymbol{z})) = [\![M]\!]^t(\lim \alpha_k(\boldsymbol{\theta}), \lim \alpha_k(\boldsymbol{z})) \\ &= [\![M]\!]^t(\lim \alpha_k(\boldsymbol{\theta}'), \lim \alpha_k(\boldsymbol{z}')) = \lim [\![M]\!]^t(\alpha_k(\boldsymbol{\theta}'), \alpha_k(\boldsymbol{z}')) = \lim \alpha_k(y')\end{aligned}$$

which demonstrates $\mathcal{A}_M, (\alpha_k)_{k\in\mathbb{N}} \not\models_{\lim} y \not\Downarrow y'$ and the goal clause is satisfied. □

8.5.3 Decidability

Finally, to prove decidability of continuity, we modify the resolution proof system by adapting the constraint refutation rule. To determine applicability of the constraint refutation rule we need to decide (the y_i, y_i' may potentially be amongst the x_j)

$$\models_{\lim} A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge y_1 \Downarrow y_1' \wedge \cdots \wedge y_k \Downarrow y_k' \wedge t_1 \not\Downarrow t_1' \wedge \cdots t_\ell \not\Downarrow t_\ell' \to \bot$$

which we abbreviate to

$$\models_{\lim} A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{y} \Downarrow \boldsymbol{y'} \wedge \boldsymbol{t} \not\Downarrow \boldsymbol{t'} \to \bot \tag{8.9}$$

The following insight is key:

Proposition 8.38. *The following statements are equivalent:*

(i) $\models_{\lim} A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{y} \Downarrow \boldsymbol{y'} \wedge \boldsymbol{t} \not\Downarrow \boldsymbol{t'} \to \bot$
(ii) $A\boldsymbol{x} \bowtie \boldsymbol{b}$ *is unsatisfiable or* $\models A\boldsymbol{x} \leq \boldsymbol{b} \wedge \boldsymbol{y} = \boldsymbol{y'} \wedge \boldsymbol{t} \neq \boldsymbol{t'} \to \bot$

Consequently, (8.9) is decidable with the randomised decision procedure of Section 8.3.1. (For this it is important that the arguments of $\Downarrow$ are affine.)

Proof. For the if-direction suppose $(\alpha_k)_{k\in\mathbb{N}}$ is a convergent sequence of valuations satisfying $\alpha_k \models A\boldsymbol{x} \bowtie \boldsymbol{b}$ for all $k \in \mathbb{N}$ and $\lim \alpha_k(y_1) = \lim \alpha_k(y_1'), \ldots, \lim \alpha_k(y_k) = \lim \alpha_k(y_k')$. Let $\alpha := \lim \alpha_k$. Observe that $\alpha \models A\boldsymbol{x} \leq \boldsymbol{b}$ and

$$\alpha(y_i) = \lim \alpha_k(y_i) = \lim \alpha_k(y_i') = \alpha(y_i')$$

Since $A\boldsymbol{x} \bowtie \boldsymbol{b}$ cannot be unsatisfiable, by assumption for some i,

$$\lim [\![t_i]\!](\alpha_k) = [\![t_i]\!](\alpha) = [\![t_i']\!](\alpha) = \lim [\![t_i']\!](\alpha_k)$$

and thus, $(\alpha_k)_{k\in\mathbb{N}} \not\models_{\lim} t_i \not\Downarrow t_i'$. Therefore, $(\alpha_k)_{k\in\mathbb{N}} \models_{\lim} A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{y} \Downarrow \boldsymbol{y'} \wedge \boldsymbol{t} \not\Downarrow \boldsymbol{t'} \to \bot$.

Conversely, suppose $A\boldsymbol{x} \bowtie \boldsymbol{b}$ is satisfiable and α is a valuation satisfying $\alpha \models A\boldsymbol{x} \leq \boldsymbol{b}$ and $\alpha(y_1) = \alpha(y_1'), \ldots, \alpha(y_k) = \alpha(y_k')$. By satisfiability, there exists α_0 satisfying $\alpha_0 \models A\boldsymbol{x} \bowtie \boldsymbol{b}$. Define $\alpha_k \coloneqq \frac{1}{k} \cdot \alpha_0 + \left(1 - \frac{1}{k}\right) \cdot \alpha$ (defined pointwise). Note that $\alpha_k \models A\boldsymbol{x} \bowtie \boldsymbol{b}$ for all $k \in \mathbb{N}$. Furthermore, by definition $(\alpha_k)_{k\in\mathbb{N}} \models_{\lim} y_i \Downarrow y_i'$. Thus, by assumption, $(\alpha_k)_{k\in\mathbb{N}} \not\models_{\lim} t_i \not\Downarrow t_i'$ for some $1 \leq i \leq \ell$, which implies

$$[\![t_i]\!](\alpha) = \lim [\![t_i]\!](\alpha_k) = \lim [\![t_i']\!](\alpha_k) = [\![t_i']\!](\alpha)$$ □

Proof System

We modify the constraint refutation rule of our proof system accordingly:

Resolution
$$\frac{B \to P\, x_1 \cdots x_n\, t' \qquad P\, t_1 \cdots t_n\, y \wedge B' \to \bot}{(B \wedge B'[t'/y])[\boldsymbol{t}/\boldsymbol{x}] \to \bot}$$

Constraint Refutation
$$\frac{A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{y} \Downarrow \boldsymbol{y'} \wedge \boldsymbol{t} \not\Downarrow \boldsymbol{t'} \to \bot}{\bot}$$

provided there exist valuations α and α' satisfying

$$\alpha \models A\boldsymbol{x} \bowtie \boldsymbol{b} \qquad \alpha' \models A\boldsymbol{x} \leq \boldsymbol{b} \wedge \boldsymbol{y} = \boldsymbol{y'} \wedge \boldsymbol{t} \neq \boldsymbol{t'}$$

By the additional constraint on goal clauses in Section 8.5.1, y does not occur in a $\Downarrow$-atom (but it may occur in a $\not\Downarrow$-atom) and the resolvent is a goal clause (in the modified sense) again.

We write $\Phi \vdash_{\lim} G$ if G can be derived by these proof rules.

Proposition 8.39 (Soundness)**.** *If $\Phi \vdash_{\lim} G$ then for all $\mathcal{A} \models_{\lim} \Phi$, $\mathcal{A} \models_{\lim} G$. In particular, if $\Phi \vdash_{\lim} \bot$ then Φ is lim-unsatisfiable.*

Proof. The soundness of the constraint refutation rule is due to Proposition 8.38.

For the resolution rule suppose

$$\mathcal{A} \models B \to P\, x_1 \cdots x_n\, t' \qquad \mathcal{A} \models_{\lim} P\, t_1 \cdots t_n\, y \wedge B' \to \bot$$

To show that also $\mathcal{A} \models_{\lim} (B \wedge B'[t'/y])[\boldsymbol{t}/\boldsymbol{x}] \to \bot$ let $(\alpha_k)_{k\in\mathbb{N}}$ be a convergent sequence of valuations satisfying $\mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} B[\boldsymbol{t}/\boldsymbol{x}]$. Since bodies of definite clauses contain neither $\Downarrow$ nor $\not\Downarrow$ this means for every $k \in \mathbb{N}$, $\mathcal{A}, \alpha_k \models B[\boldsymbol{t}/\boldsymbol{x}]$. Due to $\mathcal{A} \models B \to P\, x_1 \cdots x_n\, t'$, $\mathcal{A}, \alpha_k \models (P\, \boldsymbol{x}\, t')[\boldsymbol{t}/\boldsymbol{x}]$ must hold for every $k \in \mathbb{N}$. Consequently, by Eq. (8.1),

$$\mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \models ((P\, t_1 \cdots t_n\, y)[t'/y])[\boldsymbol{t}/\boldsymbol{x}]$$

and by

$$\mathcal{A} \models_{\lim} P\, t_1 \cdots t_n\, y \wedge B' \to \bot$$

$\mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \models_{\lim} (B'[t'/y])[\boldsymbol{t}/\boldsymbol{x}] \to \bot$ must hold, completing the proof. □

Theorem 8.40. *Let Φ be a finite set of non-recursive HoCHCs.*

Φ is lim-unsatisfiable iff $\Phi \vdash_{\lim} \bot$, and this is decidable.

We stress that as a consequence of Proposition 8.37, verifying continuity of terms $\boldsymbol{\theta} \mid \Sigma \vdash_{\text{aff}} M : R^{(a)}$ is decidable.

Proof. Recall the vector $\boldsymbol{c}_G \coloneqq (c_1, \dots, c_n) \in \mathbb{N}^n$ associated to goal clauses G from Section 8.3.2, where c_i is the number of occurrences of P_i in G, and the canonical structure $\mathcal{A}_\Phi$, which satisfies all definite clauses (cf. Section 8.4).

If Φ is lim-unsatisfiable there exists a goal clause $\Phi \vdash_{\lim} G$ such that $\mathcal{A}_\Phi \not\models_{\lim} G$ and $\boldsymbol{c}_G$ is minimal (w.r.t. the lexicographic order). This means that there exists a convergent $(\alpha_k)_{k\in\mathbb{N}}$ satisfying $\mathcal{A}, (\alpha_k)_{k\in\mathbb{N}} \not\models G$.

If $\boldsymbol{c}_G = (0, \dots, 0)$ then G is a background goal clause and therefore the constraint refutation rule is applicable by Proposition 8.38.

Otherwise, there exists a maximal P_i occurring in G. By definition of non-recursive clauses such a P_i can only occur in an atom $P_i\, t_1 \cdots t_n\, y$. Hence, G has the form

$$P_i\, N_1 \cdots N_n\, y \wedge B \to \bot$$

and $\mathcal{A}, \alpha_k \models P_i\, N_1 \cdots N_n\, y$ for all $k \in \mathbb{N}$.

Intuitively, since the sequence $(\alpha_k)_{k\in\mathbb{N}}$ is infinite and there are only finitely many clauses, there must be a definite clause responsible for an (infinite) subsequence. Formally, there exists an increasing sequence $(k_\ell)_{\ell\in\mathbb{N}}$ and

$$B' \to P_i\, x_1 \cdots x_n\, t'$$

α'_ℓ satisfying $\mathcal{A}_\Phi, \alpha'_\ell \models B'$ and $\alpha'_\ell(x_1) = \mathcal{A}_\Phi[\![t_1]\!](\alpha_{k_\ell}), \dots, \alpha'_\ell(x_n) = \mathcal{A}_\Phi[\![t_n]\!](\alpha_{k_\ell})$ and $\mathcal{A}_\Phi[\![t']\!](\alpha'_\ell) = \alpha_{k_\ell}(y)$. Consequently (assuming the variables have potentially been renamed to ensure disjointness),

$$\mathcal{A}_\Phi, (\alpha_{k_\ell} \uplus \alpha'_\ell)_{\ell\in\mathbb{N}} \models_{\lim} B'[\boldsymbol{t}/\boldsymbol{y}]$$
$$\mathcal{A}_\Phi, (\alpha_{k_\ell} \uplus \alpha'_\ell)_{\ell\in\mathbb{N}} \models_{\lim} (B[t'/y])[\boldsymbol{t}/\boldsymbol{x}]$$

(For the latter we note that $(\alpha_k)_{k\in\mathbb{N}} \models_{\lim} s \not\Downarrow s'$ iff $(\alpha_{k_\ell})_{k\in\mathbb{N}} \models_{\lim} s \not\Downarrow s'$ since $[\![s]\!], [\![s']\!]$ are continuous and $(\alpha_k)_{k\in\mathbb{N}}$ is convergent.)

Consequently, $\mathcal{A}_\Phi$ also does not satisfy

$$B'[\boldsymbol{t}/\boldsymbol{x}] \wedge (B[t'/y])[\boldsymbol{t}/\boldsymbol{x}] \to \bot$$

which is derivable in the proof system and has strictly reduced the number of occurrences of P_i (and only introduced symbols amongst $P_1, \dots, P_{i-1}$). This contradicts the minimality assumption of $\boldsymbol{c}_G$. □

With this we can also easily prove Proposition 8.33:

Proof of Proposition 8.33. If $[\![M]\!]$ is not continuous then by Proposition 8.37, Φ_M^{cont} is unsatisfiable. Hence, by completeness of the proof system (Theorem 8.40) there exists $\Phi_M^{\lim} \vdash_{\lim} A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{y} \Downarrow \boldsymbol{y}' \wedge t \not\Downarrow t' \to \bot$ such that $\not\models_{\lim} A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{y} \Downarrow \boldsymbol{y}' \wedge t \not\Downarrow t' \to \bot$. By Proposition 8.38, $\not\models A\boldsymbol{x} \leq \boldsymbol{b} \wedge \boldsymbol{y} = \boldsymbol{y}' \wedge t \neq t' \to \bot$.

Finally, the derivation of

$$\Phi_M^{\lim} \vdash_{\lim} A\boldsymbol{x} \bowtie \boldsymbol{b} \wedge \boldsymbol{y} \Downarrow \boldsymbol{y}' \wedge t \not\Downarrow t' \to \bot$$

can be modified to a derivation

$$\Phi_M^{\mathrm{c}} \vdash_{\mathrm{res}} A\boldsymbol{x} \leq \boldsymbol{b} \wedge \boldsymbol{y} = \boldsymbol{y}' \wedge t \neq t' \to \bot$$

which can be refuted. Unsatisfiability of Φ_M^{c} immediately follows from soundness of the original proof system (Theorem 8.16). □

The following example illustrates incompleteness for recursive clauses:

Example 8.41 (Incompleteness for Recursive Clauses)**.** We can axiomatise the graph of the Heaviside step function $[(-) \geq 0]$ in a rather extravagant fashion as

$$\begin{aligned} x < -1 &\to \mathrm{Gr}\, x\, 0 \\ \mathrm{Gr}\,(2x)\, y &\to \mathrm{Gr}\, x\, y \\ x \geq 0 &\to \mathrm{Gr} x\, 1 \end{aligned}$$

Continuity can be expressed as $\mathrm{Gr}\, x\, y \wedge \mathrm{Gr}\, x'\, y' \wedge x \Downarrow x' \wedge y \not\Downarrow y' \to \bot$. The clauses are not lim-satisfiable because the goal clause fails for $(\alpha_k)_{k\in\mathbb{N}}$ defined by

$$\alpha_k(x) \coloneqq -\frac{1}{k} \qquad \alpha_k(y) \coloneqq 0 \qquad \alpha_k(x') \coloneqq \frac{1}{k} \qquad \alpha_k(y') \coloneqq 1$$

However, it is easy to see that the only background goal clause we can derive via resolution (exhibiting $0 \not\Downarrow 1$ needed for a potential refutation) is

$$2^i \cdot x < -1 \wedge x' \geq 0 \wedge x \Downarrow x' \wedge 0 \not\Downarrow 1 \to \bot$$

for every $i \in \mathbb{N}$. However, $\models_{\lim} 2^i \cdot x < -1 \wedge x' \geq 0 \wedge x \Downarrow x' \wedge 0 \not\Downarrow 1 \to \bot$ and the clause cannot be refuted because there is no $(\alpha_k)_{k\in\mathbb{N}}$ satisfying $\alpha_k(x) < -2^{-i}$ and $\alpha_k(x') \geq 0$ for all $k \in \mathbb{N}$, yet $\lim_{k\to\infty} \alpha_k(x) = \lim_{k\to\infty} \alpha_k(x')$.

8.6 Conclusion and Related Work

In this chapter, we have proven that (under mild conditions) the reparameterisation gradient estimator is unbiased for continuous but non-differentiable models. Motivated by this, we have studied methods based on higher-order logic with a background theory to verify continuity. Our development has culminated in a sound and complete reduction from the verification problem of continuity to a (non-standard) satisfiability problem. Furthermore, we have devised novel randomised algorithms for deciding the latter.

Related Work

Similarly as Lemma 8.3, the concurrent development of [Lee et al., 2023, Theorem E.2] also notes in the appendix that the differentiability requirement in Lemma 2.17 can be relaxed to admit continuous but non-differentiable primitives. However, their program analysis is conservative in that whenever a guard depends on a sample, that sample cannot be reparameterised, and they do not try to establish continuity in the presence of conditionals.

[Chaudhuri et al., 2010, Chaudhuri et al., 2012, Barthe et al., 2020a] study the verification of continuity in the context of programming languages (the former two papers

for an imperative language, the latter paper for a higher-order functional language). Despite basing their approach on type systems, their rule for conditionals has a complex side condition to ensure equivalence of both branches on the boundary. [Chaudhuri et al., 2010, Chaudhuri et al., 2012] claim that these side conditions can be discharged by SMT solvers without elaborating on details. More generally, it is not clear under what circumstances typability is (efficiently) decidable.

Furthermore, whilst these papers provide soundness results, they do not make any statements about the completeness of their approaches.

Our approach to verifying continuity is more uniform and we fully benefit from a separation of concerns: we encode continuity as a (single) satisfiability problem in higher-order logic with a background theory, which we show to be decidable, and we prove the (revised) encoding to be both sound and complete.

Chapter 9

Conclusion and Future Directions

In this thesis we have studied the foundations of fast yet correct variational inference for probabilistic programming, an innovative programming paradigm to pose and automatically solve Bayesian inference problems.

We have shown that almost surely terminating programs have densities which are almost everywhere differentiable. This result is of significant practical interest because many inference algorithms are gradient-based.

Unfortunately, almost everywhere differentiability is only necessary but not sufficient for the correctness of one of the most successful inference algorithms: variational inference with the reparameterisation gradient estimator. In particular, discontinuities may cause this estimator, which is favourable in practice due to low variance, to be biased.

As a basis for our formal development, we have posed a general stochastic optimisation problem generalising variational inference and guaranteed its well-definedness by suitable assumptions and type systems.

To circumvent the bias caused by discontinuities, we have endowed our language with a smoothed semantics parameterised by an accuracy coefficient. We have formally shown that not only is the reparameterisation gradient estimator unbiased (for fixed accuracy coefficient) but also that stochastic gradient descent is correct. Furthermore, we have posed a type system guaranteeing that the smoothed objective function converges uniformly to the original, unsmoothed objective function.

Applying stochastic gradient descent to a fixed smoothing requires the tuning of the accuracy coefficient hyperparameter. As an alternative, we have proposed Diagonalisation Gradient Descent, which automatically enhances the quality of the approximation *during* the optimisation trajectory. This method enjoys strong theoretical guarantees: asymptotically a stationary point of the original, unsmoothed problem is attained.

Our experimental evaluation demonstrates important advantages over the state of the art: significantly lower variance (score estimator), unbiasedness (reparameterisation estimator), simplicity, wider applicability and lower variance (unbiased correction thereof), as well as stability over the choice of (initial) accuracy coefficients (fixed smoothing).

Finally, we have shown that the reparameterisation gradient estimator is unbiased for continuous but non-differentiable models, and we have investigated techniques based on higher-order logic to verify continuity. We have identified a suitable non-standard background theory and proposed novel randomised decision procedures. Moreover, we have presented a sound and complete encoding of continuity in the logic, and concluded decidability of the verification of continuity for a large class of models.

Future Directions

Our analyses of the optimisation algorithms are asymptotic and focus on stationary points, which leaves room for future research strengthening the guarantees (convergence rates, avoidance of saddle points, etc.).

Besides, we plan to explore methods *adaptively* tuning the accuracy coefficient rather than *a priori* fixing a scheme (as for Diagonalisation SGD). Whilst the present work was primarily concerned with theoretical guarantees, we anticipate adaptive methods to outperform fixed schemes in practice at the cost of fewer theoretical guarantees or significantly more complex correctness proofs.

A natural avenue for future research is to make the language and type systems more complete, i.e. to support more well-behaved programs. For instance, we have a devised an incomplete type system guaranteeing uniform convergence of the smoothed objective function to the original unsmoothed objective function by ensuring guard safety, i.e. that guards are almost everywhere not 0. Analogous to the verification of continuity, we plan to investigate employing methods based on higher-order logic, which do not suffer from incompleteness to the same extent, for verifying guard safety. A particularly interesting aspect is that this requires proposing a semantics suitable for reasoning about such almost-sure properties and designing decision procedures.

Furthermore, an interesting direction is to support recursion beyond Chapter 3. In particular, it is not obvious how to extend the smoothing approach in a computationally tractable way since a natural generalisation would result in possibly infinite terms. The naïve remedy, *estimating* the smoothed gradient in a Monte Carlo fashion, may result in high variance.

Furthermore, the encoding for the verification of continuity in Chapter 8 can be straightforwardly extended to accommodate recursion. However, as we have seen in Example 8.41, the proof system is incomplete in general if recursion is permitted. Therefore, it would be interesting to identify fragments for which resolution is complete, implying that logic-based methods can be employed to verify continuity soundly even in the presence of recursion.

Finally, in a recent paper, [Arya et al., 2022] present a novel reparameterisation-like estimator for discrete distributions using the new concept of *stochastic derivatives*. It would be interesting to investigate whether their techniques can be extended to general conditionals occurring in programming languages.

Bibliography

[Aiken, 1999] Aiken, A. (1999). Introduction to set constraint-based program analysis. *Sci. Comput. Program.*, 35(2):79–111.

[Arya et al., 2022] Arya, G., Schauer, M., Schäfer, F., and Rackauckas, C. (2022). Automatic differentiation of programs with discrete randomness. In *NeurIPS*.

[Aumann, 1961] Aumann, R. J. (1961). Borel structures for function spaces. *Illinois Journal of Mathematics*, 5.

[Barrett et al., 2021] Barrett, C. W., Sebastiani, R., Seshia, S. A., and Tinelli, C. (2021). Satisfiability modulo theories. In Biere, A., Heule, M., van Maaren, H., and Walsh, T., editors, *Handbook of Satisfiability - Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, pages 1267–1329. IOS Press.

[Barthe et al., 2020a] Barthe, G., Crubillé, R., Dal Lago, U., and Gavazzo, F. (2020a). On the versatility of open logical relations. In *European Symposium on Programming*, pages 56–83. Springer.

[Barthe et al., 2020b] Barthe, G., Katoen, J.-P., and Silva, A., editors (2020b). *Foundations of Probabilistic Programming*. Cambridge University Press.

[Benaïm et al., 2005] Benaïm, M., Hofbauer, J., and Sorin, S. (2005). Stochastic approximations and differential inclusions. *SIAM J. Control. Optim.*, 44(1):328–348.

[Benthem and Doets, 1983] Benthem, J. V. and Doets, K. (1983). Higher-order logic. In Gabbay, D. M. and Guenthner, F., editors, *Handbook of Philosophical Logic*, volume 164 of *Synthese Library (Studies in Epistemology, Logic, Methodology, and Philosophy of Science)*. Springer, Dordrecht.

[Bertsekas, 2015] Bertsekas, D. (2015). *Convex optimization algorithms*. Athena Scientific.

[Bertsekas, 1975] Bertsekas, D. P. (1975). *Nondifferentiable optimization via approximation*, pages 1–25. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Bertsekas and Tsitsiklis, 2000] Bertsekas, D. P. and Tsitsiklis, J. N. (2000). Gradient convergence in gradient methods with errors. *SIAM J. Optim.*, 10(3):627–642.

[Bhat et al., 2012] Bhat, S., Agarwal, A., Vuduc, R. W., and Gray, A. G. (2012). A type theory for probability density functions. In Field, J. and Hicks, M., editors, *Proceedings*

of the 39th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2012, Philadelphia, Pennsylvania, USA, January 22-28, 2012, pages 545–556. ACM.

[Bhat et al., 2017] Bhat, S., Borgström, J., Gordon, A. D., and Russo, C. V. (2017). Deriving probability density functions from probabilistic functional programs. *Logical Methods in Computer Science*, 13(2).

[Bichsel et al., 2018] Bichsel, B., Gehr, T., and Vechev, M. T. (2018). Fine-grained semantics for probabilistic programs. In Ahmed, A., editor, *Programming Languages and Systems - 27th European Symposium on Programming, ESOP 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings*, volume 10801 of *Lecture Notes in Computer Science*, pages 145–185. Springer.

[Bingham et al., 2019] Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P. A., Horsfall, P., and Goodman, N. D. (2019). Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, 20:28:1–28:6.

[Bishop, 2007] Bishop, C. M. (2007). *Pattern recognition and machine learning, 5th Edition*. Information science and statistics. Springer.

[Bjørner et al., 2015] Bjørner, N., Gurfinkel, A., McMillan, K. L., and Rybalchenko, A. (2015). Horn clause solvers for program verification. In *Fields of Logic and Computation II - Essays Dedicated to Yuri Gurevich on the Occasion of His 75th Birthday*, pages 24–51.

[Bjørner et al., 2012] Bjørner, N., McMillan, K. L., and Rybalchenko, A. (2012). Program verification as satisfiability modulo theories. In *10th International Workshop on Satisfiability Modulo Theories, SMT 2012, Manchester, UK, June 30 - July 1, 2012*, pages 3–11.

[Blake and Zisserman, 1987] Blake, A. and Zisserman, A. (1987). *Visual Reconstruction*. MIT Press.

[Blei et al., 2017] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

[Borgström et al., 2016] Borgström, J., Lago, U. D., Gordon, A. D., and Szymczak, M. (2016). A lambda-calculus foundation for universal probabilistic programming. In *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, ICFP 2016, Nara, Japan, September 18-22, 2016*, pages 33–46.

[Botev and Ridder, 2017] Botev, Z. and Ridder, A. (2017). Variance reduction. In *Wiley StatsRef: Statistics Reference Online*, pages 1–6. John Wiley & Sons, Inc.

[Brunel et al., 2020] Brunel, A., Mazza, D., and Pagani, M. (2020). Backpropagation in the simply typed lambda-calculus with linear negation. *Proc. ACM Program. Lang.*, 4(POPL):64:1–64:27.

[Castellan and Paquet, 2019] Castellan, S. and Paquet, H. (2019). Probabilistic programming inference via intensional semantics. In *European Symposium on Programming*, pages 322–349. Springer.

[Cathcart Burn et al., 2018] Cathcart Burn, T., Ong, C. L., and Ramsay, S. J. (2018). Higher-order constrained horn clauses for verification. *PACMPL*, 2(POPL):11:1–11:28.

[Cathcart Burn et al., 2021] Cathcart Burn, T., Ong, L., Ramsay, S. J., and Wagner, D. (2021). Initial limit datalog: a new extensible class of decidable constrained horn clauses. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–13. IEEE.

[Chaganty et al., 2013] Chaganty, A., Nori, A., and Rajamani, S. (2013). Efficiently sampling probabilistic programs via program analysis. In *Artificial Intelligence and Statistics*, pages 153–160.

[Chaudhuri et al., 2010] Chaudhuri, S., Gulwani, S., and Lublinerman, R. (2010). Continuity analysis of programs. In Hermenegildo, M. V. and Palsberg, J., editors, *Proceedings of the 37th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2010, Madrid, Spain, January 17-23, 2010*, pages 57–70. ACM.

[Chaudhuri et al., 2012] Chaudhuri, S., Gulwani, S., and Lublinerman, R. (2012). Continuity and robustness of programs. *Commun. ACM*, 55(8):107–115.

[Chaudhuri and Solar-Lezama, 2010] Chaudhuri, S. and Solar-Lezama, A. (2010). Smooth interpretation. In Zorn, B. G. and Aiken, A., editors, *Proceedings of the 2010 ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2010, Toronto, Ontario, Canada, June 5-10, 2010*, pages 279–291. ACM.

[Chaudhuri and Solar-Lezama, 2011] Chaudhuri, S. and Solar-Lezama, A. (2011). Smoothing a program soundly and robustly. In Gopalakrishnan, G. and Qadeer, S., editors, *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*, volume 6806 of *Lecture Notes in Computer Science*, pages 277–292. Springer.

[Clarke, 1976] Clarke, L. A. (1976). A system to generate test data and symbolically execute programs. *IEEE Trans. Software Eng.*, 2(3):215–222.

[Conway, 1973] Conway, J. B. (1973). *Functions of One Complex Variable I*. Springer-Verlag, second edition.

[Cousot and Cousot, 1977] Cousot, P. and Cousot, R. (1977). Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In Graham, R. M., Harrison, M. A., and Sethi, R., editors, *Conference Record of the Fourth ACM Symposium on Principles of Programming Languages, Los Angeles, California, USA, January 1977*, pages 238–252. ACM.

[Culpepper and Cobb, 2017] Culpepper, R. and Cobb, A. (2017). Contextual equivalence for probabilistic programs with continuous random variables and scoring. In Yang, H., editor, *Programming Languages and Systems - 26th European Symposium on Programming, ESOP 2017, Held as Part of the European Joint Conferences on Theory and*

Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings, volume 10201 of *Lecture Notes in Computer Science*, pages 368–392. Springer.

[Cusumano-Towner et al., 2019] Cusumano-Towner, M. F., Saad, F. A., Lew, A. K., and Mansinghka, V. K. (2019). Gen: a general-purpose probabilistic programming system with programmable inference. In McKinley, K. S. and Fisher, K., editors, *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2019, Phoenix, AZ, USA, June 22-26, 2019*, pages 221–236. ACM.

[Dahlqvist and Kozen, 2020] Dahlqvist, F. and Kozen, D. (2020). Semantics of higher-order probabilistic programs with conditioning. *Proc. ACM Program. Lang.*, 4(POPL):57:1–57:29.

[Davidson-Pilon, 2015] Davidson-Pilon, C. (2015). *Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference*. Addison-Wesley Professional.

[de Amorim and Lam, 2022] de Amorim, P. H. A. and Lam, C. (2022). Distribution theoretic semantics for non-smooth differentiable programming. *CoRR*, abs/2207.05946.

[Duane et al., 1987] Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid monte carlo. *Physics letters B*.

[Duchi et al., 2011] Duchi, J. C., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159.

[Ehrhard et al., 2018] Ehrhard, T., Pagani, M., and Tasson, C. (2018). Measurable cones and stable, measurable functions: a model for probabilistic higher-order programming. *PACMPL*, 2(POPL):59:1–59:28.

[Ehrhard et al., 2014] Ehrhard, T., Tasson, C., and Pagani, M. (2014). Probabilistic coherence spaces are fully abstract for probabilistic PCF. In *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014*, pages 309–320.

[Enderton, 2001] Enderton, H. B. (2001). *A Mathematical Introduction to Logic*. Academic Press, second edition.

[Faw et al., 2022] Faw, M., Tziotis, I., Caramanis, C., Mokhtari, A., Shakkottai, S., and Ward, R. (2022). The power of adaptivity in SGD: self-tuning step sizes with unbounded gradients and affine variance. In Loh, P. and Raginsky, M., editors, *Conference on Learning Theory, 2-5 July 2022, London, UK*, volume 178 of *Proceedings of Machine Learning Research*, pages 313–355. PMLR.

[Figurnov et al., 2018] Figurnov, M., Mohamed, S., and Mnih, A. (2018). Implicit reparameterization gradients. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 439–450.

[Frölicher and Kriegl, 1988] Frölicher, A. and Kriegl, A. (1988). *Linear Spaces and Differentiation Theory*. Interscience, J. Wiley and Son, New York.

[Fu, 2006] Fu, M. C. (2006). Chapter 19 gradient estimation. In Henderson, S. G. and Nelson, B. L., editors, *Simulation*, volume 13 of *Handbooks in Operations Research and Management Science*, pages 575–616. North-Holland.

[Glynn and Whitt, 1992] Glynn, P. W. and Whitt, W. (1992). The asymptotic efficiency of simulation estimators. *Oper. Res.*, 40(3):505–520.

[Gorinova et al., 2022] Gorinova, M. I., Gordon, A. D., Sutton, C., and Vákár, M. (2022). Conditional independence by typing. *ACM Trans. Program. Lang. Syst.*, 44(1):4:1–4:54.

[Gorinova et al., 2020] Gorinova, M. I., Moore, D., and Hoffman, M. D. (2020). Automatic reparameterisation of probabilistic programs. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3648–3657. PMLR.

[Grebenshchikov et al., 2012] Grebenshchikov, S., Lopes, N. P., Popeea, C., and Rybalchenko, A. (2012). Synthesizing software verifiers from proof rules. In *ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI '12, Beijing, China - June 11 - 16, 2012*, pages 405–416.

[Gupta et al., 2011] Gupta, A., Popeea, C., and Rybalchenko, A. (2011). Solving recursion-free horn clauses over LI+UIF. In Yang, H., editor, *Programming Languages and Systems - 9th Asian Symposium, APLAS 2011, Kenting, Taiwan, December 5-7, 2011. Proceedings*, volume 7078 of *Lecture Notes in Computer Science*, pages 188–203. Springer.

[Hazan et al., 2016] Hazan, E., Levy, K. Y., and Shalev-Shwartz, S. (2016). On graduated optimization for stochastic non-convex problems. In Balcan, M. and Weinberger, K. Q., editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1833–1841. JMLR.org.

[Henkin, 1950] Henkin, L. (1950). Completeness in the theory of types. *J. Symb. Log.*, 15(2):81–91.

[Heunen et al., 2017a] Heunen, C., Kammar, O., Staton, S., and Yang, H. (2017a). A convenient category for higher-order probability theory. *Proc. Symposium Logic in Computer Science*.

[Heunen et al., 2017b] Heunen, C., Kammar, O., Staton, S., and Yang, H. (2017b). A convenient category for higher-order probability theory. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12.

[Hinton, 2012] Hinton, G. (2012). Neural networks for machine learning: Lecture 6a overview of mini-batch gradient descent.

[Hoffman et al., 2013] Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. W. (2013). Stochastic variational inference. *J. Mach. Learn. Res.*, 14(1):1303–1347.

[Hörmander, 2015] Hörmander, L. (2015). *The Analysis of Linear Partial Differential Operators I: Distribution Theory and Fourier Analysis.* Classics in Mathematics. Springer Berlin Heidelberg.

[Huot et al., 2023] Huot, M., Lew, A. K., Mansinghka, V. K., and Staton, S. (2023). ωpap spaces: Reasoning denotationally about higher-order, recursive probabilistic and differentiable programs. In *LICS*, pages 1–14.

[Huot et al., 2020] Huot, M., Staton, S., and Vákár, M. (2020). Correctness of automatic differentiation via diffeologies and categorical gluing. In Goubault-Larrecq, J. and König, B., editors, *Foundations of Software Science and Computation Structures - 23rd International Conference, FOSSACS 2020, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2020, Dublin, Ireland, April 25-30, 2020, Proceedings*, volume 12077 of *Lecture Notes in Computer Science*, pages 319–338. Springer.

[Hur et al., 2015] Hur, C., Nori, A. V., Rajamani, S. K., and Samuel, S. (2015). A provably correct sampler for probabilistic programs. In *35th IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2015, December 16-18, 2015, Bangalore, India*, pages 475–488.

[Ismail and Shan, 2016] Ismail, W. M. and Shan, C. (2016). Deriving a probability density calculator (functional pearl). In Garrigue, J., Keller, G., and Sumii, E., editors, *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, ICFP 2016, Nara, Japan, September 18-22, 2016*, pages 47–59. ACM.

[Jang et al., 2017] Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[Jankowiak and Obermeyer, 2018] Jankowiak, M. and Obermeyer, F. (2018). Pathwise derivatives beyond the reparameterization trick. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2240–2249. PMLR.

[Jochems et al., 2023] Jochems, J., Jones, E., and Ramsay, S. J. (2023). Higher-order MSL horn constraints. *Proc. ACM Program. Lang.*, 7(POPL):2017–2047.

[Kallenberg, 2002] Kallenberg, O. (2002). *Foundations of modern probability.* Probability and its Applications (New York). Springer-Verlag, New York, second edition.

[Kaminski et al., 2019] Kaminski, B. L., Katoen, J., and Matheja, C. (2019). On the hardness of analyzing probabilistic programs. *Acta Inf.*, 56(3):255–285.

[Khajwal, 2022] Khajwal, B. (2022). An empirical evaluation of novel gradient estimators for discontinuous probabilistic models via smoothing. Master's thesis, University of Oxford.

[Khajwal et al., 2023] Khajwal, B., Ong, C. L., and Wagner, D. (2023). Fast and correct gradient-based optimisation for probabilistic programming via smoothing. In Wies, T., editor, *Programming Languages and Systems - 32nd European Symposium on Programming, ESOP 2023, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2023, Paris, France, April 22-27, 2023, Proceedings*, volume 13990 of *Lecture Notes in Computer Science*, pages 479–506. Springer.

[King, 1976] King, J. C. (1976). Symbolic execution and program testing. *Commun. ACM*, 19(7):385–394.

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[Kingma and Welling, 2014] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

[Kiselyov, 2016] Kiselyov, O. (2016). Problems of the Lightweight Implementation of Probabilistic Programming. In *PPS Workshop*.

[Klenke, 2014] Klenke, A. (2014). *Probability Theory: A Comprehensive Course*. Universitext. Springer London.

[Kozen, 1979] Kozen, D. (1979). Semantics of probabilistic programs. In *20th Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 29-31 October 1979*, pages 101–114.

[Kroening and Strichman, 2008] Kroening, D. and Strichman, O. (2008). *Decision Procedures - An Algorithmic Point of View*. Texts in Theoretical Computer Science. An EATCS Series. Springer.

[Kucukelbir et al., 2015] Kucukelbir, A., Ranganath, R., Gelman, A., and Blei, D. M. (2015). Automatic variational inference in stan. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 568–576.

[Lee, 2009] Lee, J. M. (2009). *Manifolds and Differential Geometry*, volume 107 of *Graduate Studies in Mathematics*. AMS.

[Lee, 2013] Lee, J. M. (2013). *An introduction to smooth manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer, second edition.

[Lee et al., 2023] Lee, W., Rival, X., and Yang, H. (2023). Smoothness analysis for probabilistic programs with application to optimised variational inference. *Proc. ACM Program. Lang.*, 7(POPL):335–366.

[Lee et al., 2020a] Lee, W., Yu, H., Rival, X., and Yang, H. (2020a). On correctness of automatic differentiation for non-differentiable functions. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information*

Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

[Lee et al., 2020b] Lee, W., Yu, H., Rival, X., and Yang, H. (2020b). Towards verified stochastic variational inference for probabilistic programs. *PACMPL*, 4(POPL).

[Lee et al., 2018] Lee, W., Yu, H., and Yang, H. (2018). Reparameterization gradient for non-differentiable models. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 5558–5568.

[Leivant, 1994] Leivant, D. (1994). Higher order logic. In Gabbay, D. M., Hogger, C. J., and Robinson, J. A., editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, pages 229–321. Oxford University Press, Inc., New York, NY, USA.

[Lew et al., 2019] Lew, A. K., Cusumano-Towner, M. F., Sherman, B., Carbin, M., and Mansinghka, V. K. (2019). Trace types and denotational semantics for sound programmable inference in probabilistic languages. *Proceedings of the ACM on Programming Languages*, 4(POPL):1–32.

[Lew et al., 2020] Lew, A. K., Cusumano-Towner, M. F., Sherman, B., Carbin, M., and Mansinghka, V. K. (2020). Trace types and denotational semantics for sound programmable inference in probabilistic languages. *Proc. ACM Program. Lang.*, 4(POPL):19:1–19:32.

[Lew et al., 2023] Lew, A. K., Huot, M., Staton, S., and Mansinghka, V. K. (2023). ADEV: sound automatic differentiation of expected values of probabilistic programs. *Proc. ACM Program. Lang.*, 7(POPL):121–153.

[Maddison et al., 2017] Maddison, C. J., Mnih, A., and Teh, Y. W. (2017). The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[Mahajan et al., 2012] Mahajan, M., Nimbhorkar, P., and Varadarajan, K. R. (2012). The planar k-means problem is np-hard. *Theor. Comput. Sci.*, 442:13–21.

[Mak et al., 2021] Mak, C., Ong, C. L., Paquet, H., and Wagner, D. (2021). Densities of almost surely terminating probabilistic programs are differentiable almost everywhere. In Yoshida, N., editor, *Programming Languages and Systems - 30th European Symposium on Programming, ESOP 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 - April 1, 2021, Proceedings*, volume 12648 of *Lecture Notes in Computer Science*, pages 432–461. Springer.

[Mazza and Pagani, 2021] Mazza, D. and Pagani, M. (2021). Automatic differentiation in PCF. *Proc. ACM Program. Lang.*, 5(POPL).

[Metropolis and Ulam, 1949] Metropolis, N. and Ulam, S. (1949). The monte carlo method. *J. Am. Stat. Assoc.*, 44:335.

[Minh and Gregor, 2014] Minh, A. and Gregor, K. (2014). Neural variational inference and learning in belief networks. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1791–1799. JMLR.org.

[Mityagin, 2015] Mityagin, B. (2015). The zero set of a real analytic function.

[Mohamed et al., 2020] Mohamed, S., Rosca, M., Figurnov, M., and Mnih, A. (2020). Monte carlo gradient estimation in machine learning. *J. Mach. Learn. Res.*, 21:132:1–132:62.

[Moreau and Aeyels, 2000] Moreau, L. and Aeyels, D. (2000). Optimization of discontinuous functions: a generalised theory of differentiation. *SIAM J. OPTIM.*, 11(1):53–69.

[Munkres, 1999] Munkres, J. R. (1999). *Topology*. Prentice Hall, New Delhi,, 2nd. edition.

[Murphy, 2012] Murphy, K. P. (2012). *Machine Learning: A Probabilististic Perspective*. MIT Press.

[Neal, 2011] Neal, R. M. (2011). Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, page 113.

[Ong and Wagner, 2019] Ong, C. L. and Wagner, D. (2019). Hochc: A refutationally complete and semantically invariant system of higher-order logic modulo theories. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, pages 1–14.

[Opper and Saad, 2001] Opper, M. and Saad, D. (2001). *Advanced Mean Field Methods: Theory and Practice*. The MIT Press.

[Ranganath et al., 2014] Ranganath, R., Gerrish, S., and Blei, D. M. (2014). Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, pages 814–822.

[Reddi et al., 2018] Reddi, S. J., Kale, S., and Kumar, S. (2018). On the convergence of adam and beyond. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

[Reed and Simon, 2003] Reed, M. and Simon, B. (2003). *Methods of Modern Mathematical Physics: Functional analysis. I.* World Published Corporation.

[Rezende et al., 2014] Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. JMLR.org.

[Robbins and Monro, 1951] Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

[Rudin, 1976] Rudin, W. (1976). *Principles of Mathematical Analysis*. International Series in Pure and Applied Mathematics. McGraw-Hill Education, 3rd edition edition.

[Ruiz et al., 2016] Ruiz, F. J. R., Titsias, M. K., and Blei, D. M. (2016). The generalized reparameterization gradient. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 460–468.

[Rümmer et al., 2015] Rümmer, P., Hojjat, H., and Kuncak, V. (2015). On recursion-free horn clauses and craig interpolation. *Formal Methods Syst. Des.*, 47(1):1–25.

[Saheb-Djahromi, 1978] Saheb-Djahromi, N. (1978). Probabilistic lcf. In *International Symposium on Mathematical Foundations of Computer Science*, pages 442–451. Springer.

[Schulman et al., 2015] Schulman, J., Heess, N., Weber, T., and Abbeel, P. (2015). Gradient estimation using stochastic computation graphs. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3528–3536.

[Ścibior et al., 2017] Ścibior, A., Kammar, O., Vákár, M., Staton, S., Yang, H., Cai, Y., Ostermann, K., Moss, S. K., Heunen, C., and Ghahramani, Z. (2017). Denotational validation of higher-order bayesian inference. *Proceedings of the ACM on Programming Languages*, 2(POPL):60.

[Scott, 1993] Scott, D. S. (1993). A type-theoretical alternative to ISWIM, CUCH, OWHY. *Theor. Comput. Sci.*, 121(1&2):411–440.

[Shumway and Stoffer, 2005] Shumway, R. H. and Stoffer, D. S. (2005). *Time Series Analysis and Its Applications*. Springer Texts in Statistics. Springer-Verlag.

[Sieber, 1990] Sieber, K. (1990). Relating full abstraction results for different programming languages. In *Foundations of Software Technology and Theoretical Computer Science, Tenth Conference, Bangalore, India, December 17-19, 1990, Proceedings*, pages 373–387.

[Soudjani et al., 2017] Soudjani, S. E. Z., Majumdar, R., and Nagapetyan, T. (2017). Multilevel monte carlo method for statistical model checking of hybrid systems. In Bertrand, N. and Bortolussi, L., editors, *Quantitative Evaluation of Systems - 14th International Conference, QEST 2017, Berlin, Germany, September 5-7, 2017, Proceedings*, volume 10503 of *Lecture Notes in Computer Science*, pages 351–367. Springer.

[Stacey, 2011] Stacey, A. (2011). Comparative smootheology. *Theory and Applications of Categories*, 25(4):64–117.

[Staton, 2017] Staton, S. (2017). Commutative semantics for probabilistic programming. In *Programming Languages and Systems - 26th European Symposium on Programming, ESOP 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, pages 855–879.

[Staton et al., 2016] Staton, S., Yang, H., Wood, F. D., Heunen, C., and Kammar, O. (2016). Semantics for probabilistic programming: higher-order functions, continuous distributions, and soft constraints. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 525–534.

[Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. The MIT Press, second edition.

[Titsias and Lázaro-Gredilla, 2014] Titsias, M. K. and Lázaro-Gredilla, M. (2014). Doubly stochastic variational bayes for non-conjugate inference. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1971–1979.

[Tu, 2011] Tu, L. W. (2011). *An introduction to manifolds*. Universitext. Springer-Verlag.

[Tucker et al., 2017] Tucker, G., Mnih, A., Maddison, C. J., Lawson, D., and Sohl-Dickstein, J. (2017). REBAR: low-variance, unbiased gradient estimates for discrete latent variable models. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2627–2636.

[Unno and Terauchi, 2015] Unno, H. and Terauchi, T. (2015). Inferring simple solutions to recursion-free horn clauses via sampling. In Baier, C. and Tinelli, C., editors, *Tools and Algorithms for the Construction and Analysis of Systems - 21st International Conference, TACAS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, volume 9035 of *Lecture Notes in Computer Science*, pages 149–163. Springer.

[Vákár et al., 2019] Vákár, M., Kammar, O., and Staton, S. (2019). A domain theory for statistical probabilistic programming. *PACMPL*, 3(POPL):36:1–36:29.

[van de Meent et al., 2018] van de Meent, J., Paige, B., Yang, H., and Wood, F. (2018). An introduction to probabilistic programming. *CoRR*, abs/1809.10756.

[Wand et al., 2018] Wand, M., Culpepper, R., Giannakopoulos, T., and Cobb, A. (2018). Contextual equivalence for a probabilistic language with continuous random variables and recursion. *PACMPL*, 2(ICFP):87:1–87:30.

[Wheeden et al., 1977] Wheeden, R., Wheeden, R., and Zygmund, A. (1977). *Measure and Integral: An Introduction to Real Analysis*. Chapman & Hall/CRC Pure and Applied Mathematics. Taylor & Francis.

[Wingate et al., 2011] Wingate, D., Stuhlmüller, A., and Goodman, N. D. (2011). Lightweight implementations of probabilistic programming languages via transformational compilation. In Gordon, G. J., Dunson, D. B., and Dudík, M., editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pages 770–778. JMLR.org.

[Wingate and Weber, 2013] Wingate, D. and Weber, T. (2013). Automated variational inference in probabilistic programming. *CoRR*, abs/1301.1299.

[Xu et al., 2019] Xu, M., Quiroz, M., Kohn, R., and Sisson, S. A. (2019). Variance reduction properties of the reparameterization trick. In Chaudhuri, K. and Sugiyama, M., editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 2711–2720. PMLR.

[Yang, 2019] Yang, H. (2019). Some semantic issues in probabilistic programming languages (invited talk). In Geuvers, H., editor, *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany*, volume 131 of *LIPIcs*, pages 4:1–4:6. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

[Zang, 1981] Zang, I. (1981). Discontinuous optimization by smoothing. *Mathematics of Operations Research*, 6(1):140–152.

[Zhang et al., 2019] Zhang, C., Butepage, J., Kjellstrom, H., and Mandt, S. (2019). Advances in Variational Inference. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):2008–2026.

[Zhou et al., 2019] Zhou, Y., Gram-Hansen, B. J., Kohn, T., Rainforth, T., Yang, H., and Wood, F. (2019). LF-PPL: A low-level first order probabilistic programming language for non-differentiable models. In Chaudhuri, K. and Sugiyama, M., editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 148–157. PMLR.

[Zhou et al., 2020] Zhou, Y., Yang, H., Teh, Y. W., and Rainforth, T. (2020). Divide, conquer, and combine: a new inference strategy for probabilistic programs with stochastic support. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11534–11545. PMLR.

Appendix A

Supplementary Materials for Chapter 3

A.1 More Details on Example 3.2

We prove that Op_1 and Op_2 of Example 3.2 satisfy Assumption 3.1.

(i) Clearly, all constant and projection functions are analytic. Since analytic functions are total and differentiable (hence continuous) functions, they are Borel-measurable. Therefore, due to the fact that analytic functions are closed under pairs and composition [Conway, 1973, Prop. 2.4], it remains to check whether the boundary of $f^{-1}([0,\infty))$ has measure zero.

Since $f^{-1}(\mathring{A}) \subseteq \mathring{f^{-1}}(A)$ and $\overline{f^{-1}(A)} \subseteq f^{-1}(\overline{A})$ for any subset $A \subseteq \mathbb{R}$, we have $\partial f^{-1}(A) \subseteq f^{-1}(\overline{A}) \setminus f^{-1}(\mathring{A}) = f^{-1}(\partial A)$. Letting $A = [0,\infty)$, we have

$$\partial f^{-1}([0,\infty)) \subseteq f^{-1}(\{0\})$$

Applying the well-known result [Mityagin, 2015] that the zero set of all analytic functions, except the zero function, has measure zero we conclude that $\partial f^{-1}([0,\infty))$ has measure zero. It is easy to see that if f is the zero function,

$$\partial f^{-1}([0,\infty)) = \partial \mathbb{R}^n = \varnothing$$

has measure zero.

(ii) Clearly all constant functions and projections are in Op_2.

Note that the set of finite unions of (possibly unbounded) rectangles forms an algebra $\mathcal{A}$ (i.e. a collection of subsets of $\mathbb{R}^n$ closed under complements and finite unions, hence finite intersections). Then $\operatorname{dom} f = f^{-1}(-\infty, 0] \cup f^{-1}(0,\infty) \in \mathcal{A}$. Besides, for every $U \in \mathcal{A}$, $\mathrm{Leb}(\partial U) = 0$ (because $\mathrm{Leb}(\partial R) = 0$ for every rectangle R).

It remains to prove that Op_2 is closed under composition. For this, suppose that $f : \mathbb{R}^\ell \rightharpoonup \mathbb{R} \in \mathrm{Op}_2$ and $g_1, \ldots, g_\ell : \mathbb{R}^m \rightharpoonup \mathbb{R} \in \mathrm{Op}_2$. It is straightforward to see that $\operatorname{dom} f \circ \langle g_i \rangle_{i=1}^{\ell}$ remains open (note that $\circ$ is relational composition). Moreover, for every x in $\operatorname{dom} f \circ \langle g_i \rangle_{i=1}^{\ell}$, $\langle g_i \rangle_{i=1}^{\ell}(x)$ is an interior point in $\operatorname{dom} f$ which is open.

It follows that the (standard) chain rule is applicable, and we have $f \circ \langle g_i \rangle_{i=1}^{\ell}$ is differentiable at x. Besides, suppose I is a (possibly unbounded) interval. By assumption there are $m \in \mathbb{N}$ and (potentially unbounded) intervals $I_{i,j}$, where $1 \leq i \leq m$ and $1 \leq j \leq \ell$ such that $f^{-1}(I) = \bigcup_{i=1}^{m} I_{i,1} \times \cdots \times I_{i,\ell}$. Observe that

$$\begin{aligned}
&\left(f \circ \langle g_i \rangle_{i=1}^{\ell}\right)^{-1}(I) \\
&= \left\{\boldsymbol{r} \in \operatorname{dom} g_1 \cap \cdots \cap \operatorname{dom} g_\ell \mid (g_1(\boldsymbol{r}), \ldots, g_\ell(\boldsymbol{r})) \in f^{-1}(I)\right\} \\
&= \bigcup_{i=1}^{m} \left\{\boldsymbol{r} \in \operatorname{dom} g_1 \cap \cdots \cap \operatorname{dom} g_\ell \mid g_1(\boldsymbol{r}) \in I_{i,1} \wedge \cdots \wedge g_\ell(\boldsymbol{r}) \in I_{i,\ell}\right\} \\
&= \bigcup_{i=1}^{m} g_1^{-1}(I_{i,1}) \cap \cdots \cap g_\ell^{-1}(I_{i,\ell})
\end{aligned}$$

and this is in $\mathcal{A}$ because algebras are closed under finite unions and intersections.

A.2 Supplementary Materials for Section 3.2

A.2.1 Supplementary Materials for Section 3.2.1

The following substitution property holds for symbolic terms $\mathcal{M}$ and $\mathcal{N}$:

$$\operatorname{dom} \lfloor \mathcal{M}[\mathcal{N}/y] \rfloor \subseteq \operatorname{dom} \lfloor \mathcal{M} \rfloor \cap \operatorname{dom} \lfloor \mathcal{N} \rfloor \tag{A.1}$$

$$\lfloor \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s})[\lfloor \mathcal{N} \rfloor (\boldsymbol{r}, \boldsymbol{s})/y] \equiv \lfloor \mathcal{M}[\mathcal{N}/y] \rfloor (\boldsymbol{r}, \boldsymbol{s}) \tag{A.2}$$

Lemma 3.8. *Suppose the set* Op *of primitives satisfies Assumption* 3.1(i).

(i) *For each symbolic value $\mathcal{V}$ of type R, by identifying* $\operatorname{dom} \|\mathcal{V}\|$ *with a subset of* $\mathbb{R}^{m+n}$, *we have* $\|\mathcal{V}\| \in \mathrm{Op}$.

(ii) *If* Op *also satisfies Assumption* 3.1(ii) *then for each symbolic term $\mathcal{M}$,*

$$\lfloor \mathcal{M} \rfloor : \mathbb{R}^m \times \mathbb{S}_n \rightharpoonup \Lambda$$

is differentiable in the interior of its domain.

Proof. (i) We prove the first part by induction on symbolic values.

- For $r' \in \mathbb{R}$, $\|\underline{r'}\|$ is a constant function and $\|x_i\|$ and $\|\alpha_j\|$ are projections, which are in Op by assumption.
- Next, suppose $\mathcal{V}$ is a symbolic value $\boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell)$. By the inductive hypothesis, each $\|\mathcal{V}_i\| \in \mathrm{Op}$. It suffices to note that

$$\left\|\boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell)\right\| (\boldsymbol{r}, \boldsymbol{s}) = f(\|\mathcal{V}_1\| (\boldsymbol{r}, \boldsymbol{s}), \ldots, \|\mathcal{V}_\ell\| (\boldsymbol{r}, \boldsymbol{s}))$$

  for $(\boldsymbol{r}, \boldsymbol{s}) \in \operatorname{dom} \boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell)$. Therefore, $\left\|\boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell)\right\|$ because Op is assumed to be closed under composition.
- Finally, note that we do not need to consider abstractions because they do not have type R.

(ii)
- Note that $\lfloor x_i \rfloor$ and $\lfloor \alpha_j \rfloor$ are projection functions and $\lfloor \underline{r} \rfloor$ are constant functions, which are (everywhere) differentiable functions.

– Besides, $\boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell)$ is a symbolic value and on its domain,

$$\left\lfloor \boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell) \right\rfloor = \mathbb{\lambda}(\boldsymbol{r}, \boldsymbol{s}).\, \underline{\left\| \boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell) \right\| (\boldsymbol{r}, \boldsymbol{s})}$$

$\left\| \boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell) \right\|$ is in Op by the first part and by assumption this implies differentiability.

– The function $\lfloor \lambda y.\mathcal{M} \rfloor$ is obtained by composing $\lfloor \mathcal{M} \rfloor$ with the function $\Lambda \to \Lambda$ mapping $L \mapsto \lambda y.L$ The latter is easily seen to be differentiable: recall that $\Lambda = \bigcup_{n \in \mathbb{N}} \bigcup_M \{M\} \times \mathbb{R}^n$, where M ranges over skeleton terms with n placeholders. On each component $\{M\} \times \mathbb{R}^n$ the function acts as $(M, \boldsymbol{x}) \mapsto (\lambda y.M, \boldsymbol{x})$; it is simply one of the coproduct injections, hence differentiable.

– The cases of $\lfloor \mathbf{Y}\mathcal{M} \rfloor$ and $\lfloor \mathbf{score}(\mathcal{M}) \rfloor$ are analogous.

– The function $\lfloor MN \rfloor$ is obtained by composing $\lfloor M \rfloor \times \lfloor N \rfloor$ with the diagonal map $(\boldsymbol{r}, \boldsymbol{s}) \mapsto ((\boldsymbol{r}, \boldsymbol{s}), (\boldsymbol{r}, \boldsymbol{s}))$; both are differentiable.

– The cases of $\left\lfloor \underline{f}(\mathcal{M}_1, \ldots, \mathcal{M}_\ell) \right\rfloor$ and $\lfloor \mathbf{if}\ \mathcal{L} < 0\ \mathbf{then}\ \mathcal{M}\ \mathbf{else}\ \mathcal{N} \rfloor$ are similar, using diagonal maps of different arities.

– The function $\lfloor \mathbf{sample} \rfloor$ is a constant function, so it is differentiable. This covers all cases. □

A.2.2 Supplementary Materials for Section 3.2.2

Lemma 3.11 (Subject Construction)**.** *Let $\mathcal{M}$ be a symbolic term.*

(i) *If $\mathcal{M}$ is a symbolic value then for all symbolic contexts $\mathcal{E}$ and symbolic redexes $\mathcal{R}$, $\mathcal{M} \not\equiv \mathcal{E}[\mathcal{R}]$.*
(ii) *If $\mathcal{M} \equiv \mathcal{E}_1[\mathcal{R}_1] \equiv \mathcal{E}_2[\mathcal{R}_2]$ then $\mathcal{E}_1 \equiv \mathcal{E}_2$ and $\mathcal{R}_1 \equiv \mathcal{R}_2$.*
(iii) *If $\mathcal{M}$ is not a symbolic value and $\operatorname{dom} \lfloor \mathcal{M} \rfloor \neq \emptyset$ then there exist $\mathcal{E}$ and $\mathcal{R}$ such that $\mathcal{M} \equiv \mathcal{E}[\mathcal{R}]$.*

Proof. We prove all parts of the lemma simultaneously by structural induction on $\mathcal{M}$.

– First, note that for every $\mathcal{E}$ and $\mathcal{R}$ all of the following holds

$$x_i \not\equiv \mathcal{E}[\mathcal{R}] \qquad \alpha_j \not\equiv \mathcal{E}[\mathcal{R}] \qquad y \not\equiv \mathcal{E}[\mathcal{R}] \qquad \underline{r} \not\equiv \mathcal{E}[\mathcal{R}] \qquad \lambda y.\, \mathcal{M} \not\equiv \mathcal{E}[\mathcal{R}]$$

and the left hand sides are symbolic values.

– Note that $\operatorname{dom} \left\lfloor \boxed{f}(\mathcal{M}_1, \ldots, \mathcal{M}_\ell) \right\rfloor = \emptyset$ unless it is a symbolic value. Besides, for every $\mathcal{E}$ and $\mathcal{R}$, $\boxed{f}(\mathcal{M}_1, \ldots, \mathcal{M}_\ell) \not\equiv \mathcal{E}[\mathcal{R}]$.

– If $\mathcal{M} \equiv \mathcal{N}_1\, \mathcal{N}_2$ then $\mathcal{M}$ is not a symbolic value.
Suppose that $\mathcal{N}_1$ is an abstraction. If $\mathcal{N}_2$ is a symbolic value then $\mathcal{M}$ is a symbolic redex and by the first part of the inductive hypothesis, $\mathcal{M} \equiv \mathcal{E}[\mathcal{R}]$ implies $\mathcal{E} \equiv []$ and $\mathcal{R} \equiv \mathcal{M}$.
If $\mathcal{M}_2$ is not a symbolic value then $\mathcal{M}$ is not a symbolic redex. Note that $\mathcal{M} \equiv \mathcal{E}[\mathcal{R}]$ implies $\mathcal{E} \equiv \mathcal{N}_1\, \mathcal{E}'$. By the second part of the inductive hypothesis $\mathcal{E}'$ and $\mathcal{R}$ are unique if they exist. Besides, due to $\operatorname{dom} \lfloor \mathcal{M} \rfloor \subseteq \operatorname{dom} \lfloor \mathcal{N}_2 \rfloor$ and the third part of the inductive hypothesis, such $\mathcal{E}'$ and $\mathcal{R}$ exist if $\operatorname{dom} \lfloor \mathcal{M} \rfloor \neq \emptyset$.
If $\mathcal{N}_1$ is not an abstraction it cannot be a symbolic value and $\mathcal{M} \equiv \mathcal{E}[\mathcal{R}]$ implies $\mathcal{E} \equiv \mathcal{E}'\, \mathcal{N}_2$. By the second part of the inductive hypothesis, $\mathcal{E}'$ and $\mathcal{R}$ are unique if they exist. Besides, because of $\operatorname{dom} \lfloor \mathcal{M} \rfloor \subseteq \operatorname{dom} \lfloor \mathcal{M}_1 \rfloor$ and the third part of the inductive hypothesis, such $\mathcal{E}'$ and $\mathcal{R}$ exists if $\operatorname{dom} \lfloor \mathcal{M} \rfloor \neq \emptyset$.

- Next, suppose $\mathcal{M} \equiv \underline{f}(\mathcal{N}_1, \ldots, \mathcal{N}_\ell)$, which is clearly not a symbolic value.
  If all $\mathcal{N}_i$ are symbolic values, $\mathcal{M}$ is a symbolic redex and by the first part of the inductive hypothesis, $\mathcal{E} \equiv []$ and $\mathcal{R} \equiv \mathcal{M}$ are unique such that $\mathcal{M} \equiv \mathcal{E}[\mathcal{R}]$.
  Otherwise, suppose i is minimal such that $\mathcal{N}_i$ is not a symbolic value. Clearly, $\mathcal{M} \equiv \mathcal{E}[\mathcal{R}]$ implies $\mathcal{E} \equiv \underline{f}(\mathcal{N}_1, \ldots, \mathcal{N}_{i-1}, \mathcal{E}', \mathcal{N}_{i+1}, \ldots, \mathcal{N}_\ell)$ and $\mathcal{N}_i \equiv \mathcal{E}'[\mathcal{R}]$. By the second part of the inductive hypothesis $\mathcal{E}'$ and $\mathcal{R}$ are unique if they exist. Besides due to $\operatorname{dom}\lfloor\mathcal{M}\rfloor \subseteq \operatorname{dom}\lfloor\mathcal{N}_i\rfloor$ and the third part of the inductive hypothesis such $\mathcal{E}'$ and $\mathcal{R}$ exist if $\operatorname{dom}\lfloor\mathcal{M}\rfloor \neq \emptyset$.
- If $\mathcal{M} \equiv \mathbf{Y}\mathcal{N}$, $\mathcal{M} \equiv \mathbf{sample}$ or $\mathcal{M} \equiv \mathbf{score}(\mathcal{N})$, which are not symbolic values, then this is obvious (using the inductive hypothesis).
- Finally, suppose $\mathcal{M} \equiv (\mathbf{if}\ \mathcal{L} < 0\ \mathbf{then}\ \mathcal{N}_1\ \mathbf{else}\ \mathcal{N}_2) \equiv \mathcal{E}_1[\mathcal{R}_1] \equiv \mathcal{E}_2[\mathcal{R}_2]$. If $\mathcal{L}$ is a symbolic value then $\mathcal{M}$ is a symbolic redex and by the first part of the inductive hypothesis, $\mathcal{M} \equiv \mathcal{E}[\mathcal{R}]$ implies $\mathcal{E} \equiv []$ and $\mathcal{R} \equiv \mathcal{M}$.
  If $\mathcal{L}$ is not a symbolic value then $\mathcal{M} \equiv \mathcal{E}[\mathcal{R}]$ implies $\mathcal{E} \equiv \mathbf{if}\ \mathcal{E}' < 0\ \mathbf{then}\ \mathcal{N}_1\ \mathbf{else}\ \mathcal{N}_2$ and $\mathcal{L} \equiv \mathcal{E}'[\mathcal{R}]$. By the second part of the inductive $\mathcal{E}'$ and $\mathcal{R}$ are unique if they exist. Due to $\operatorname{dom}\lfloor\mathcal{M}\rfloor \subseteq \operatorname{dom}\lfloor\mathcal{L}\rfloor$ and the third part of the inductive hypothesis such $\mathcal{E}'$ and $\mathcal{R}$ exist provided that $\operatorname{dom}\lfloor\mathcal{M}\rfloor \neq \emptyset$. □

We obtain that for all $\mathcal{E}$, $\mathcal{M}$ and $(\boldsymbol{r}, \boldsymbol{s}) \in \operatorname{dom}\lfloor\mathcal{E}[\mathcal{M}]\rfloor$:

$$\lfloor\mathcal{E}\rfloor(\boldsymbol{r}, \boldsymbol{s})\,[\lfloor\mathcal{M}\rfloor(\boldsymbol{r}, \boldsymbol{s})] \equiv \lfloor\mathcal{E}[\mathcal{M}]\rfloor(\boldsymbol{r}, \boldsymbol{s}) \tag{A.3}$$

Lemma A.1. *Suppose* $\langle\!\langle \mathcal{R}, w, U\rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{R}', w', U'\rangle\!\rangle$, $(\boldsymbol{r}, \boldsymbol{s}) \in U$ *and* $(\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) \in U'$. *Then* $\langle\lfloor\mathcal{R}\rfloor(\boldsymbol{r}, \boldsymbol{s}), w(\boldsymbol{r}, \boldsymbol{s}), \boldsymbol{s}\rangle \to \langle\lfloor\mathcal{R}'\rfloor(\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}), w'(\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}), \boldsymbol{s} +\!\!+ \boldsymbol{s'}\rangle$.

Proof. We prove the lemma by case analysis on the symbolic redex contractions.

- First, suppose $\langle\!\langle \mathbf{sample}, w, U\rangle\!\rangle \Rightarrow \langle\!\langle \alpha_{n+1}, w', U'\rangle\!\rangle$. Note that $s' = [r']$ for some $0 < r' < 1$. Then

  $$\langle \mathbf{sample}, w(\boldsymbol{r}, \boldsymbol{s}), \boldsymbol{s}\rangle \to \left\langle \underline{r'}, w(\boldsymbol{r}, \boldsymbol{s}), \boldsymbol{s} +\!\!+ [r']\right\rangle$$

  and $w(\boldsymbol{r}, \boldsymbol{s}) = w'(\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'})$ and $\lfloor\alpha_{n+1}\rfloor(\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) \equiv \underline{r'}$.
- Suppose $\langle\!\langle \mathbf{score}(\mathcal{V}), w, U\rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{V}, w \cdot \|\mathcal{V}\|, U \cap \|\mathcal{V}\|^{-1}[0, \infty)\rangle\!\rangle$. Then

  $$(\boldsymbol{r}, \boldsymbol{s}) = (\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) \in U \cap \|\mathcal{V}\|^{-1}[0, \infty)$$

  Hence, there must exist $r' \geq 0$ such that $\lfloor\mathcal{V}\rfloor(\boldsymbol{r}, \boldsymbol{s}) \equiv \lfloor\mathcal{V}\rfloor(\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) \equiv \underline{r'}$. Besides,

  $$\langle \mathbf{score}(\lfloor\mathcal{V}\rfloor(\boldsymbol{r}, \boldsymbol{s})), w(\boldsymbol{r}, \boldsymbol{s}), \boldsymbol{s}\rangle \to \left\langle \underline{r'}, w(\boldsymbol{r}, \boldsymbol{s}) \cdot r', \boldsymbol{s}\right\rangle$$

  and $(w \cdot \|\mathcal{V}\|)(\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) = w(\boldsymbol{r}, \boldsymbol{s}) \cdot r'$.
- Suppose $\langle\!\langle \mathbf{if}\ \mathcal{V} < 0\ \mathbf{then}\ \mathcal{M}\ \mathbf{else}\ \mathcal{N}, w, U\rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{M}, w, U \cap \|\mathcal{V}\|^{-1}(-\infty, 0)\rangle\!\rangle$. Note that $(\boldsymbol{r}, \boldsymbol{s}) = (\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) \in U \cap \|\mathcal{V}\|^{-1}(-\infty, 0)$. Thus, $\|\mathcal{V}\|(\boldsymbol{r}, \boldsymbol{s}) < 0$. Therefore,

  $$\begin{aligned}&\left\langle \mathbf{if}\ \underline{\|\mathcal{V}\|(\boldsymbol{r}, \boldsymbol{s})} < 0\ \mathbf{then}\ \lfloor\mathcal{M}\rfloor(\boldsymbol{r}, \boldsymbol{s})\ \mathbf{else}\ \lfloor\mathcal{N}\rfloor(\boldsymbol{r}, \boldsymbol{s}), w(\boldsymbol{r}, \boldsymbol{s}), \boldsymbol{s}\right\rangle \\ &\to \langle\lfloor\mathcal{M}\rfloor(\boldsymbol{r}, \boldsymbol{s}), w(\boldsymbol{r}, \boldsymbol{s}), \boldsymbol{s}\rangle\end{aligned}$$

- Similar for the else-branch.

– Suppose $\langle\!\langle (\lambda y.\, \mathcal{M})\, \mathcal{V}, w, U \rangle\!\rangle \to \langle\!\langle \mathcal{M}[\mathcal{V}/y], w, U \rangle\!\rangle$. Then $(\boldsymbol{r}, \boldsymbol{s}) = (\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) \in U$. By Lemma 3.7, $\lfloor \mathcal{V} \rfloor\, (\boldsymbol{r}, \boldsymbol{s})$ is a value. Hence,

$$\langle (\lambda y.\, \lfloor \mathcal{M} \rfloor\, (\boldsymbol{r}, \boldsymbol{s}))\ \lfloor \mathcal{V} \rfloor\, (\boldsymbol{r}, \boldsymbol{s}), w(\boldsymbol{r}, \boldsymbol{s}), \boldsymbol{s} \rangle \to \langle (\lfloor \mathcal{M} \rfloor\, (\boldsymbol{r}, \boldsymbol{s}))[\lfloor \mathcal{V} \rfloor\, (\boldsymbol{r}, \boldsymbol{s})/y], w(\boldsymbol{r}, \boldsymbol{s}), \boldsymbol{s} \rangle$$

Besides, by Eq. (A.1), $(\lfloor \mathcal{M} \rfloor\, (\boldsymbol{r}, \boldsymbol{s}))[\lfloor \mathcal{V} \rfloor\, (\boldsymbol{r}, \boldsymbol{s})/y] \equiv \lfloor \mathcal{M}[\mathcal{V}/y] \rfloor\, (\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'})$.

– Suppose $\langle\!\langle \underline{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell), w, U \rangle\!\rangle \Rightarrow \langle\!\langle \boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell), w, U \cap \mathrm{dom} \left\| \underline{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell) \right\| \rangle\!\rangle$. Then $(\boldsymbol{r}, \boldsymbol{s}) = (\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) \in U \cap \mathrm{dom} \left\| \underline{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell) \right\|$. In particular, $(\boldsymbol{r}, \boldsymbol{s}) \in \mathrm{dom}\, \|\mathcal{V}_i\|$ for each $1 \leq i \leq \ell$ and $(\|\mathcal{V}_1\|\, (\boldsymbol{r}, \boldsymbol{s}), \ldots \|\mathcal{V}_\ell\|\, (\boldsymbol{r}, \boldsymbol{s})) \in \mathrm{dom}\, f$. Therefore,

$$\begin{aligned} &\left\langle \underline{f(\|\mathcal{V}_1\|\, (\boldsymbol{r}, \boldsymbol{s}), \ldots, \|\mathcal{V}_\ell\|\, (\boldsymbol{r}, \boldsymbol{s}))}, w(\boldsymbol{r}, \boldsymbol{s}), \boldsymbol{s} \right\rangle \\ &\to \left\langle \underline{f(\|\mathcal{V}_1\|\, (\boldsymbol{r}, \boldsymbol{s}), \ldots, \|\mathcal{V}_\ell\|\, (\boldsymbol{r}, \boldsymbol{s}))}, w(\boldsymbol{r}, \boldsymbol{s}), \boldsymbol{s} \right\rangle \end{aligned}$$

because $\underline{f(\|\mathcal{V}_1\|\, (\boldsymbol{r}, \boldsymbol{s}), \ldots, \|\mathcal{V}_\ell\|\, (\boldsymbol{r}, \boldsymbol{s}))} \equiv \left\lfloor \boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell) \right\rfloor (\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'})$.

– Finally, suppose

$$\langle\!\langle \mathbf{Y}(\lambda x.\, \mathcal{M}), w, U \rangle\!\rangle \Rightarrow \langle\!\langle \lambda y.\, \mathcal{M}\, [\mathbf{Y}(\lambda x.\, \mathcal{M})/x]\, y, w, U \rangle\!\rangle$$

Then $(\boldsymbol{r}, \boldsymbol{s}) = (\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) \in U$. It holds

$$\begin{aligned} &\langle \mathbf{Y}(\lambda x.\, \lfloor \mathcal{M} \rfloor\, (\boldsymbol{r}, \boldsymbol{s})), w(\boldsymbol{r}, \boldsymbol{s}), \boldsymbol{s} \rangle \\ &\to \langle \lambda y.\, \lfloor \mathcal{M} \rfloor\, (\boldsymbol{r}, \boldsymbol{s})[\mathbf{Y}(\lambda x.\, \lfloor \mathcal{M} \rfloor\, (\boldsymbol{r}, \boldsymbol{s}))/x]\, y, w(\boldsymbol{r}, \boldsymbol{s}), \boldsymbol{s} \rangle \end{aligned}$$

and by the Substitution Eq. (A.1),

$$\lambda y.\, \lfloor \mathcal{M} \rfloor\, (\boldsymbol{r}, \boldsymbol{s})[\mathbf{Y}(\lambda x.\, \lfloor \mathcal{M} \rfloor\, (\boldsymbol{r}, \boldsymbol{s}))/x]\, y \equiv \lfloor \lambda y.\, \mathcal{M}[\mathbf{Y}(\lambda x. \mathcal{M})/x]\, y \rfloor\, (\boldsymbol{r}, \boldsymbol{s} +\!\!+ \boldsymbol{s'}) \quad \square$$

Lemma A.2. *Suppose* $\lfloor \mathcal{M} \rfloor\, (\boldsymbol{r}, \boldsymbol{s}) \equiv R$, $(\boldsymbol{r}, \boldsymbol{s}) \in U$, $w(\boldsymbol{r}, \boldsymbol{s}) = w$ *and* $\langle R, w, \boldsymbol{s} \rangle \to \langle R', w', \boldsymbol{s'} \rangle$. *Then there exists* $\langle\!\langle \mathcal{M}, w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{R}', w', U' \rangle\!\rangle$ *such that* $\lfloor \mathcal{R}' \rfloor\, (\boldsymbol{r}, \boldsymbol{s'}) \equiv R'$, $w'(\boldsymbol{r}, \boldsymbol{s'}) = w'$ *and* $(\boldsymbol{r}, \boldsymbol{s'}) \in U'$.

Proof. We prove the lemma by a case distinction on the redex contractions.

– First, suppose

$$\langle \mathbf{sample}, w, \boldsymbol{s} \rangle \to \langle \underline{r}, w, \boldsymbol{s} +\!\!+ [r] \rangle$$

where $0 < r < 1$ and $(\boldsymbol{r}, \boldsymbol{s}) \in U \subseteq \mathbb{R}^m \times \mathbb{S}_n$. Then $\langle\!\langle \mathbf{sample}, w, U \rangle\!\rangle \Rightarrow \langle\!\langle \alpha_{n+1}, w', U' \rangle\!\rangle$, where $U' = \{(\boldsymbol{r}, \boldsymbol{s} +\!\!+ [r']) \mid (\boldsymbol{r}, \boldsymbol{s}) \in U \wedge 0 < r' < 1\} \ni (\boldsymbol{r}, \boldsymbol{s} +\!\!+ [r])$ as well as $w'(\boldsymbol{r}, \boldsymbol{s} +\!\!+ [r]) = w(\boldsymbol{r}, \boldsymbol{s}) = w$. By definition, $\lfloor \alpha_{n+1} \rfloor\, (\boldsymbol{r}, \boldsymbol{s} +\!\!+ [r]) \equiv \underline{r}$.

– Suppose $\langle \mathbf{score}(\underline{r'}), w, \boldsymbol{s} \rangle \to \langle \underline{r'}, w', \boldsymbol{s} \rangle$, where $r' \geq 0$, $(\boldsymbol{r}, \boldsymbol{s}) \in U$. Then $\mathcal{M} \equiv \mathbf{score}(\mathcal{V})$ for some $\mathcal{V}$ satisfying $\lfloor \mathcal{V} \rfloor\, (\boldsymbol{r}, \boldsymbol{s}) \equiv \underline{r'}$. Hence, $\|\mathcal{V}\|\, (\boldsymbol{r}, \boldsymbol{s}) = r' \geq 0$ and

$$\langle\!\langle \mathbf{score}(\mathcal{V}), w, U \rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{V}, w \cdot \|\mathcal{V}\|, U' \rangle\!\rangle$$

where $U' := U \cap \|\mathcal{V}\|^{-1}\, [0, \infty) \ni (\boldsymbol{r}, \boldsymbol{s})$ and $(w \cdot \|\mathcal{V}\|)(\boldsymbol{r}, \boldsymbol{s}) = w \times r'$.

– Suppose $\langle \mathbf{if}\ \underline{r'} < 0\ \mathbf{then}\ M\ \mathbf{else}\ N, w, \boldsymbol{s}\rangle \to \langle M, w, \boldsymbol{s}\rangle$ because $r' < 0$ and $(\boldsymbol{r}, \boldsymbol{s}) \in U$. Suppose that $\mathcal{V}, \mathcal{M}, \mathcal{N}$ are such that $\lfloor \mathcal{V} \rfloor (\boldsymbol{r}, \boldsymbol{s}) \equiv \underline{r'}$, $\lfloor \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s}) \equiv M$ and $\lfloor \mathcal{N} \rfloor (\boldsymbol{r}, \boldsymbol{s}) \equiv N$. Then $\|\mathcal{V}\| (\boldsymbol{r}, \boldsymbol{s}) = r' < 0$ and $\langle\!\langle \mathbf{if}\ \mathcal{V} < 0\ \mathbf{then}\ \mathcal{M}\ \mathbf{else}\ \mathcal{N}, w, U\rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{M}, w, U'\rangle\!\rangle$, where $U' \coloneqq (U \cap \|\mathcal{V}\|^{-1} (-\infty, 0)) \ni (\boldsymbol{r}, \boldsymbol{s})$ by assumption.

– Similar for the else-branch.

– Suppose $\left\langle \underline{f}(r'_1, \ldots, r'_\ell), w, \boldsymbol{s}\right\rangle \to \left\langle \underline{f(r'_1, \ldots, r'_\ell)}, w, \boldsymbol{s}\right\rangle$, where $(r'_1, \ldots, r'_\ell) \in \operatorname{dom} f$. Suppose further that $\mathcal{V}_1, \ldots, \mathcal{V}_\ell$ are such that for each $1 \le i \le \ell$, $\lfloor \mathcal{V}_i \rfloor (\boldsymbol{r}, \boldsymbol{s}) \equiv \underline{r'_i}$. Since $\operatorname{dom} \left\|\underline{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell)\right\| = \operatorname{dom} \left\lfloor \underline{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell) \right\rfloor$, then we have

$$\langle\!\langle \underline{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell), w, U\rangle\!\rangle \Rightarrow \langle\!\langle \boxed{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell), w, U'\rangle\!\rangle$$

where $U' \coloneqq (U \cap \operatorname{dom} \left\|\underline{f}(\mathcal{V}_1, \ldots, \mathcal{V}_\ell)\right\|) \ni (\boldsymbol{r}, \boldsymbol{s})$.

– Suppose $\langle (\lambda y.\, M)\, V, w, \boldsymbol{s}\rangle \to \langle M[V/y], w, \boldsymbol{s}\rangle$ and $(\boldsymbol{r}, \boldsymbol{s}) \in U$. Let $\mathcal{M}, \mathcal{N}$ be such that $\lfloor \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s}) \equiv M$ and $\lfloor \mathcal{N} \rfloor (\boldsymbol{r}, \boldsymbol{s}) \equiv V$. By the Substitution Eq. (A.1),

$$\lfloor \mathcal{M}[\mathcal{N}/y] \rfloor (\boldsymbol{r}, \boldsymbol{s}) \equiv M[N/y]$$

and by Lemma 3.7, $\mathcal{N}$ must be a symbolic value. Thus,

$$\langle\!\langle (\lambda y.\, \mathcal{M})\, \mathcal{N}, w, U\rangle\!\rangle \Rightarrow \langle\!\langle \mathcal{M}[\mathcal{N}/y], w, U\rangle\!\rangle$$

– Suppose $\langle \mathbf{Y}(\lambda y.M), w, \boldsymbol{s}\rangle \to \langle \lambda z.M[\mathbf{Y}(\lambda y.M)/y]z, w, \boldsymbol{s}\rangle$ and $(\boldsymbol{r}, \boldsymbol{s}) \in U$. Let $\mathcal{M}$ be such that $\lfloor \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s}) \equiv M$, then by Substitution Eq. (A.1), we have

$$\lfloor \lambda z.\mathcal{M}[\mathbf{Y}(\lambda y.\mathcal{M})/y]z \rfloor (\boldsymbol{r}, \boldsymbol{s}) \equiv \lambda z.M[\mathbf{Y}(\lambda y.M)/y]z$$

Thus we conclude that

$$\langle\!\langle \mathbf{Y}(\lambda y.\mathcal{M}), w, U\rangle\!\rangle \Rightarrow \langle\!\langle \lambda z.\mathcal{M}[\mathbf{Y}(\lambda y.\mathcal{M})/y]z, w, U\rangle\!\rangle \qquad \square$$

Definition A.3. We extend $\lfloor \cdot \rfloor$ to symbolic contexts with domains

$$\begin{aligned}
\operatorname{dom} \lfloor [] \rfloor &\coloneqq \mathbb{R}^m \times \mathbb{S}_n \\
\operatorname{dom} \lfloor \mathcal{E}\, \mathcal{M} \rfloor &\coloneqq \operatorname{dom} \lfloor (\lambda y.\, \mathcal{M})\, \mathcal{E} \rfloor \coloneqq \operatorname{dom} \lfloor \mathcal{E} \rfloor \cap \operatorname{dom} \lfloor \mathcal{M} \rfloor \\
&\operatorname{dom} \left\lfloor \underline{f}(\mathcal{V}_1, \ldots, \mathcal{V}_{\ell-1}, \mathcal{E}, \mathcal{M}_{\ell+1}, \ldots, \mathcal{M}_n) \right\rfloor \\
&\coloneqq \operatorname{dom} \lfloor \mathcal{V}_1 \rfloor \cap \cdots \cap \operatorname{dom} \lfloor \mathcal{V}_{\ell-1} \rfloor \cap \operatorname{dom} \lfloor \mathcal{E} \rfloor \cap \operatorname{dom} \lfloor \mathcal{M}_{\ell+1} \rfloor \cap \cdots \cap \operatorname{dom} \lfloor \mathcal{M}_n \rfloor \\
\operatorname{dom} \lfloor \mathbf{Y}\mathcal{E} \rfloor &\coloneqq \operatorname{dom} \lfloor \mathbf{score}(\mathcal{E}) \rfloor \coloneqq \operatorname{dom} \lfloor \mathcal{E} \rfloor \\
\operatorname{dom} \lfloor \mathbf{if}\ \mathcal{E} < 0\ \mathbf{then}\ \mathcal{M}\ \mathbf{else}\ \mathcal{N} \rfloor &\coloneqq \operatorname{dom} \lfloor \mathcal{E} \rfloor \cap \operatorname{dom} \lfloor \mathcal{M} \rfloor \cap \operatorname{dom} \lfloor \mathcal{N} \rfloor
\end{aligned}$$

by

$$\begin{aligned}
\lfloor [] \rfloor (\boldsymbol{r}, \boldsymbol{s}) &\coloneqq [] \\
\lfloor \mathcal{E}\, \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s}) &\coloneqq (\lfloor \mathcal{E} \rfloor (\boldsymbol{r}, \boldsymbol{s}))\, (\lfloor \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s})) \\
\lfloor (\lambda y.\, \mathcal{M})\, \mathcal{E} \rfloor (\boldsymbol{r}, \boldsymbol{s}) &\coloneqq (\lambda y.\, \lfloor \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s}))\, (\lfloor \mathcal{E} \rfloor (\boldsymbol{r}, \boldsymbol{s})) \\
&\left\lfloor \underline{f}(\mathcal{V}_1, \ldots, \mathcal{V}_{\ell-1}, \mathcal{E}, \mathcal{M}_{\ell+1}, \ldots, \mathcal{M}_n) \right\rfloor (\boldsymbol{r}, \boldsymbol{s}) \\
&\coloneqq \underline{f}(\lfloor \mathcal{V}_1 \rfloor (\boldsymbol{r}, \boldsymbol{s}), \ldots, \lfloor \mathcal{V}_{\ell-1} \rfloor (\boldsymbol{r}, \boldsymbol{s}), \lfloor \mathcal{E} \rfloor (\boldsymbol{r}, \boldsymbol{s}), \lfloor \mathcal{M}_{\ell+1} \rfloor (\boldsymbol{r}, \boldsymbol{s}), \ldots, \lfloor \mathcal{M}_n \rfloor (\boldsymbol{r}, \boldsymbol{s})) \\
\lfloor \mathbf{Y}\mathcal{E} \rfloor (\boldsymbol{r}, \boldsymbol{s}) &\coloneqq \mathbf{Y}(\, \lfloor \mathcal{E} \rfloor (\boldsymbol{r}, \boldsymbol{s})) \\
\lfloor \mathbf{if}\ \mathcal{E} < 0\ \mathbf{then}\ \mathcal{M}\ \mathbf{else}\ \mathcal{N} \rfloor (\boldsymbol{r}, \boldsymbol{s}) &\coloneqq \mathbf{if}\ (\lfloor \mathcal{E} \rfloor (\boldsymbol{r}, \boldsymbol{s})) < 0\ \mathbf{then}\ (\lfloor \mathcal{M} \rfloor (\boldsymbol{r}, \boldsymbol{s}))\ \mathbf{else}\ (\lfloor \mathcal{N} \rfloor (\boldsymbol{r}, \boldsymbol{s})) \\
\lfloor \mathbf{score}(\mathcal{E}) \rfloor (\boldsymbol{r}, \boldsymbol{s}) &\coloneqq \mathbf{score}(()\, \lfloor \mathcal{E} \rfloor (\boldsymbol{r}, \boldsymbol{s}))
\end{aligned}$$

A.3 Supplementary Materials for Section Section 3.3

Lemma A.4. *Let (X, Σ_X, μ) and (Y, Σ_Y, ν) be σ-finite measure spaces. Suppose that $U \in \Sigma_X$ and that for every $r \in X$, $V_r \in \Sigma_Y$, and $W \coloneqq \{(r, s) \in X \times Y \mid s \in V_r\}$ is measurable.*

If $\mu(X \setminus U) = 0$ and for every $r \in U$, $\nu(V_r) = 0$ then $(\mu \times \nu)(W) = 0$.

Proof. Let $X_n \in \Sigma_X$ and $Y_n \in \Sigma_Y$ (for $n \in \mathbb{N}$) be such that $X = \bigcup_{n\in\mathbb{N}} X_n = X$, $Y = \bigcup_{n\in\mathbb{N}} Y_n$ and $\mu(X_n) = \nu(Y_n) < \infty$ for every $n \in \mathbb{N}$. Define $W_n \coloneqq W \cap (X_n \times Y_n)$. Clearly $(\mu \times \nu)(W_n)$ is finite.

By assumption the characteristic function $\mathbb{1}_W : X \times Y \to \mathbb{R}_{\geq 0}$ is measurable. By Fubini's theorem [Kallenberg, 2002, Thm. 1.27], for every $n \in \mathbb{N}$,

$$\mu(W_n) = \int_{X_n \times Y_n} (\mathrm{d}(\mu \times \nu))\mathbb{1}_W = \int_{X_n} (\mathrm{d}\mu) \int_{Y_n} (\mathrm{d}\nu)\mathbb{1}_W = \int_{U \cap X_n} (\mathrm{d}\mu)\lambda r.\, \nu(V_r) = 0$$

The third equation is due to $\mu(X_n \setminus U) = 0$. The claim is immediate by $W = \bigcup_{n\in\mathbb{N}} W_n$. □

Appendix B

Supplementary Materials for Chapter 5

B.1 Supplementary Materials for Section 5.1.3

To demonstrate Proposition 5.6, we posit an (infinitary) logical relation $(f, f', f_\eta, f'_\eta) \in \mathcal{R}_\tau^{U,\boldsymbol{\varphi}}$ where

- $U \subseteq \mathbb{R}^n$
- $\boldsymbol{\varphi}_{(-)} : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}^n$ is a polynomial and a strong diffeomorphism,
- $f, f' : \boldsymbol{\Theta} \times \mathbb{R}^n \to [\![\tau]\!]$ (in **QBS**)
- $f_\eta, f'_\eta : \boldsymbol{\Theta} \times \mathbb{R}^n \to [\![\tau]\!]_\eta$ (in **Fr**) for $\eta > 0$

$(f, f', f_\eta, f'_\eta) \in \mathcal{R}_\tau^{U,\boldsymbol{\varphi}}$ is defined inductively by:

(i) $(f, f', f_\eta, f'_\eta) \in \mathcal{R}_\iota^{U,\boldsymbol{\varphi}}$ if $f(\boldsymbol{\theta}, \boldsymbol{s}) = f'(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))$ and $f_\eta(\boldsymbol{\theta}, \boldsymbol{s}) = f'_\eta(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))$ for all $(\boldsymbol{\theta}, \boldsymbol{s}) \in \boldsymbol{\Theta} \times \mathbb{R}^n$ and $\eta > 0$.

(ii) $(f, f', f_\eta, f'_\eta) \in \mathcal{R}_{\tau_1 \bullet \Sigma_3 \to \tau_2}^{U,\boldsymbol{\varphi}}$ iff for all $U^{(2)}$ and $\boldsymbol{\varphi}^{(2)}$ and $(g, g', g_\eta, g'_\eta) \in \mathcal{R}_{\tau_1}^{U \times U^{(2)}, \Phi +\!\!+ \boldsymbol{\varphi}^{(2)}}$, there exists $\boldsymbol{\varphi}^{(3)}$ such that

$$(f \odot g, f' \odot g', f_\eta \odot g_\eta, f'_\eta \odot g'_\eta) \in \mathcal{R}_{\tau_2}^{U \times U^{(2)} \times \mathrm{supp}(\Sigma_3), \boldsymbol{\varphi} +\!\!+ \boldsymbol{\varphi}^{(2)} +\!\!+ \boldsymbol{\varphi}^{(3)}}$$

where again for $f : \mathbb{R}^{n_1} \to [\![\tau_1 \bullet \Sigma_3 \to \tau_2]\!]$ and $g : \mathbb{R}^{n_1+n_2} \to [\![\tau_1]\!]$ we define

$$\begin{aligned} f \odot g : \mathbb{R}^{n_1+n_2+|\Sigma_3|} &\to [\![\tau_2]\!] \\ \boldsymbol{s}_1 +\!\!+ \boldsymbol{s}_2 +\!\!+ \boldsymbol{s}_3 &\mapsto f(\boldsymbol{s}_1)(g(\boldsymbol{s}_1 +\!\!+ \boldsymbol{s}_2), \boldsymbol{s}_3) \end{aligned}$$

and use an overloaded notation in **Fr**. Furthermore for $\boldsymbol{\varphi}_{(-)} : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}^n$ and $\boldsymbol{\varphi}'_{(-)} : \boldsymbol{\Theta} \times \mathbb{R}^{n'} \to \mathbb{R}^{n'}$ we use the notation

$$\begin{aligned} \boldsymbol{\varphi} +\!\!+ \boldsymbol{\varphi}' : \boldsymbol{\Theta} \times \mathbb{R}^{n+n'} &\to \mathbb{R}^{n+n'} \\ (\boldsymbol{\theta}, \boldsymbol{s} +\!\!+ \boldsymbol{s}') &\mapsto \varphi_{\boldsymbol{\theta}}(\boldsymbol{s}) +\!\!+ \varphi_{\boldsymbol{\theta}}(\boldsymbol{s}') \end{aligned}$$

Proposition 5.6 follows immediately from the following:

Lemma B.1 (Fundamental). *If* $\theta_1 : \iota_1, \dots, \theta_m : \iota_m, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma \vdash_{\mathrm{tr}} M : \tau$ *and* $(\xi^{(1)}, \xi'^{(1)}, \xi_\eta^{(1)}, \xi'^{(1)}_\eta) \in \mathcal{R}^{\boldsymbol{\varphi}}_{\tau_1}, \dots, (\xi^{(\ell)}, \xi'^{(\ell)}, \xi_\eta^{(\ell)}, \xi'^{(\ell)}_\eta) \in \mathcal{R}^{U,\boldsymbol{\varphi}}_{\tau_\ell}$ *then there exists* $\boldsymbol{\varphi}'$ *satisfying*

$$\begin{aligned}&(\llbracket M \rrbracket * \langle \xi^{(1)}, \dots, \xi^{(\ell)} \rangle, \llbracket M \rrbracket^t * \langle \xi'^{(1)}, \dots, \xi'^{(\ell)} \rangle,\\ &\quad \llbracket M \rrbracket_\eta * \langle \xi_\eta^{(1)}, \dots, \xi_\eta^{(\ell)} \rangle, \llbracket M \rrbracket^t_\eta * \langle \xi'^{(1)}_\eta, \dots, \xi'^{(\ell)}_\eta \rangle) \in \mathcal{R}^{U \times \mathrm{supp}(\Sigma), \boldsymbol{\varphi} +\!\!+ \boldsymbol{\varphi}'}_\tau\end{aligned}$$

where $(\llbracket M \rrbracket * \langle \xi^{(1)}, \dots, \xi^{(\ell)} \rangle)(\boldsymbol{\theta}, \boldsymbol{s} +\!\!+ \boldsymbol{s}') \coloneqq \llbracket M \rrbracket(\boldsymbol{\theta}, (\xi^{(1)}(\boldsymbol{\theta}, \boldsymbol{s}), \dots, \xi^{(\ell)}(\boldsymbol{\theta}, \boldsymbol{s})), \boldsymbol{s}')$ *and similarly for* $\llbracket M \rrbracket^t$, $\llbracket M \rrbracket_\eta$ *and* $\llbracket M \rrbracket^t_\eta$.

Proof. The claim is proven by a straightforward induction on the typing derivation. We focus on the two interesting cases:

– For $\theta_1 : \iota_1, \dots, \theta_m : \iota_m, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma \vdash_{\mathrm{tr}} \mathbf{sample}\, \mathcal{D} \rhd \psi_{\boldsymbol{\theta}} : R$ by definition

$$\begin{aligned}(\llbracket \mathbf{sample}\, \mathcal{D} \rhd \psi_{\boldsymbol{\theta}} \rrbracket * \langle \xi^{(1)}, \dots, \xi^{(\ell)} \rangle)(\boldsymbol{\theta}, \boldsymbol{s} +\!\!+ [s']) &= \psi_{\boldsymbol{\theta}}(s')\\ (\llbracket \mathbf{sample}\, \mathcal{D} \rhd \psi_{\boldsymbol{\theta}} \rrbracket^t * \langle \xi'^{(1)}, \dots, \xi'^{(\ell)} \rangle)(\boldsymbol{\theta}, \boldsymbol{z} +\!\!+ [z']) &= z'\\ (\llbracket \mathbf{sample}\, \mathcal{D} \rhd \psi_{\boldsymbol{\theta}} \rrbracket_\eta * \langle \xi_\eta^{(1)}, \dots, \xi_\eta^{(\ell)} \rangle)(\boldsymbol{\theta}, \boldsymbol{s} +\!\!+ [s']) &= \psi_{\boldsymbol{\theta}}(s')\\ (\llbracket \mathbf{sample}\, \mathcal{D} \rhd \psi_{\boldsymbol{\theta}} \rrbracket^t_\eta * \langle \xi'^{(1)}_\eta, \dots, \xi'^{(\ell)}_\eta \rangle)(\boldsymbol{\theta}, \boldsymbol{z} +\!\!+ [z']) &= z'\end{aligned}$$

Furthermore, necessarily, $\psi_{(-)} \in \mathrm{Diff}_{\boldsymbol{\Theta}}(\mathrm{supp}(\mathcal{D}))$. Therefore,

$$\begin{aligned}&(\llbracket \mathbf{sample}\, \mathcal{D} \rhd \psi_{\boldsymbol{\theta}} \rrbracket * \langle \xi^{(1)}, \dots, \xi^{(\ell)} \rangle, \llbracket \mathbf{sample}\, \mathcal{D} \rhd \psi_{\boldsymbol{\theta}} \rrbracket^t * \langle \xi'^{(1)}, \dots, \xi'^{(\ell)} \rangle,\\ &\quad \llbracket \mathbf{sample}\, \mathcal{D} \rhd \psi_{\boldsymbol{\theta}} \rrbracket_\eta * \langle \xi_\eta^{(1)}, \dots, \xi_\eta^{(\ell)} \rangle,\\ &\quad \llbracket \mathbf{sample}\, \mathcal{D} \rhd \psi_{\boldsymbol{\theta}} \rrbracket^t_\eta * \langle \xi'_1, \dots, \xi'_\ell \rangle) \in \mathcal{R}^{U \times \mathrm{supp}(\mathcal{D}), \boldsymbol{\varphi} +\!\!+ [\psi]}_R\end{aligned}$$

– Suppose

$$\theta_1 : \iota_1, \dots, \theta_m : \iota_m, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma_1 +\!\!+ \Sigma_2 +\!\!+ \Sigma_3 \vdash_{\mathrm{tr}} \mathbf{if}\, L < 0\, \mathbf{then}\, M\, \mathbf{else}\, N : \iota$$

because

$$\begin{aligned}\theta_1 : \iota_1, \dots, \theta_m : \iota_m, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma_1 \vdash_{\mathrm{tr}} L : R\\ \theta_1 : \iota_1, \dots, \theta_m : \iota_m, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma_2 \vdash_{\mathrm{tr}} M : \iota\\ \theta_1 : \iota_1, \dots, \theta_m : \iota_m, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma_3 \vdash_{\mathrm{tr}} N : \iota\end{aligned}$$

By the inductive hypothesis, there exist polynomials $\boldsymbol{\varphi}^{(1)}, \boldsymbol{\varphi}^{(2)}, \boldsymbol{\varphi}^{(3)}$ which are strong diffeomorphisms, as well as

$$\begin{aligned}(\llbracket L \rrbracket * \langle \xi^{(1)}, \dots, \xi^{(\ell)} \rangle)(\boldsymbol{\theta}, \boldsymbol{s} +\!\!+ \boldsymbol{s}_1) &= (\llbracket L \rrbracket^t * \langle \xi'^{(1)}, \dots, \xi'^{(\ell)} \rangle)(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}) +\!\!+ \boldsymbol{\varphi}^{(1)}_{\boldsymbol{\theta}}(\boldsymbol{s}_1))\\ (\llbracket L \rrbracket_\eta * \langle \xi_\eta^{(1)}, \dots, \xi_\eta^{(\ell)} \rangle)(\boldsymbol{\theta}, \boldsymbol{s} +\!\!+ \boldsymbol{s}_1) &= (\llbracket L \rrbracket^t_\eta * \langle \xi'^{(1)}_\eta, \dots, \xi'^{(\ell)}_\eta \rangle)(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}) +\!\!+ \boldsymbol{\varphi}^{(1)}_{\boldsymbol{\theta}}(\boldsymbol{s}_1))\end{aligned}$$

and similarly for M and N. Consequently, if

$$(\llbracket L \rrbracket * \langle \xi^{(1)}, \dots, \xi^{(\ell)} \rangle)(\boldsymbol{\theta}, \boldsymbol{s} +\!\!+ \boldsymbol{s}_1) < 0 \tag{B.1}$$

then

$$\begin{aligned}
&(\llbracket \mathbf{if}\ L < 0\ \mathbf{then}\ M\ \mathbf{else}\ N \rrbracket * \langle \xi^{(1)}, \dots, \xi^{(\ell)} \rangle)(\boldsymbol{\theta}, \boldsymbol{s} +\!\!+ \boldsymbol{s}_1 +\!\!+ \boldsymbol{s}_2 +\!\!+ \boldsymbol{s}_3) \\
&= (\llbracket M \rrbracket * \langle \xi^{(1)}, \dots, \xi^{(\ell)} \rangle)(\boldsymbol{\theta}, \boldsymbol{s} +\!\!+ \boldsymbol{s}_2) \\
&= (\llbracket M \rrbracket^t * \langle \xi'^{(1)}, \dots, \xi'^{(\ell)} \rangle)(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}) +\!\!+ \boldsymbol{\varphi}^{(2)}_{\boldsymbol{\theta}}(\boldsymbol{s}_2)) \\
&= (\llbracket \mathbf{if}\ L < 0\ \mathbf{then}\ M\ \mathbf{else}\ N \rrbracket^t * \langle \xi'^{(1)}, \dots, \xi'^{(\ell)} \rangle) \\
&\qquad\qquad (\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}) +\!\!+ \boldsymbol{\varphi}^{(1)}_{\boldsymbol{\theta}}(\boldsymbol{s}_1) +\!\!+ \boldsymbol{\varphi}^{(2)}_{\boldsymbol{\theta}}(\boldsymbol{s}_2) +\!\!+ \boldsymbol{\varphi}^{(3)}_{\boldsymbol{\theta}}(\boldsymbol{s}_3))
\end{aligned}$$

Similarly, we can argue for the case Eq. (B.1) does not hold as well as for the requirement concerning $\llbracket \mathbf{if}\ L < 0\ \mathbf{then}\ M\ \mathbf{else}\ N \rrbracket_\eta$ and $\llbracket \mathbf{if}\ L < 0\ \mathbf{then}\ M\ \mathbf{else}\ N \rrbracket^t_\eta$. Consequently,

$$\begin{aligned}
&(\llbracket \mathbf{if}\ L < 0\ \mathbf{then}\ M\ \mathbf{else}\ N \rrbracket * \langle \xi^{(1)}, \dots, \xi^{(\ell)} \rangle, \\
&\quad \llbracket \mathbf{if}\ L < 0\ \mathbf{then}\ M\ \mathbf{else}\ N \rrbracket^t * \langle \xi'^{(1)}, \dots, \xi'^{(\ell)} \rangle, \\
&\quad \llbracket \mathbf{if}\ L < 0\ \mathbf{then}\ M\ \mathbf{else}\ N \rrbracket_\eta * \langle \xi^{(1)}_\eta, \dots, \xi^{(\ell)}_\eta \rangle, \\
&\quad \llbracket \mathbf{if}\ L < 0\ \mathbf{then}\ M\ \mathbf{else}\ N \rrbracket^t_\eta * \langle \xi'_1, \dots, \xi'_\ell \rangle) \\
&\qquad\qquad \in \mathcal{R}_\iota^{U \times \mathrm{supp}(\Sigma_1) +\!\!+ \mathrm{supp}(\Sigma_2) +\!\!+ \mathrm{supp}(\Sigma_3), \boldsymbol{\varphi} +\!\!+ \boldsymbol{\varphi}^{(1)} +\!\!+ \boldsymbol{\varphi}^{(2)} +\!\!+ \boldsymbol{\varphi}^{(3)}}
\end{aligned}$$

□

B.2 Supplementary Materials for Section 5.3.2

Lemma 5.30 (Fundamental)**.** *If* $\theta_1 : \iota_1^{(\mathbf{f},\emptyset)}, \dots, \theta_m : \iota_m^{(\mathbf{f},\emptyset)}, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma \vdash_{\mathrm{unif}} M : \tau$, $n \in \mathbb{N}$, $\mathrm{dict} : \{1, \dots, n\} \to \mathrm{Dep}$, $\xi^{(1)} \in \mathcal{DGS}_{\tau_1}^{(n,\mathrm{dict})}, \dots, \xi^{(\ell)} \in \mathcal{DGS}_{\tau_\ell}^{(n,\mathrm{dict})}$ *then*

$$\llbracket M \rrbracket^t * \langle \xi^{(1)}, \dots, \xi^{(\ell)} \rangle \in \mathcal{DGS}_\tau^{(n+|\Sigma|,\mathrm{dict}')}$$

for some $\mathrm{dict}' : \mathbb{R}^{n+|\Sigma|} \to \mathbb{R}$ *extending* dict*, where*

$$\left(\llbracket M \rrbracket^t * \langle \xi^{(1)}, \dots, \xi^{(\ell)} \rangle\right)(\boldsymbol{\theta}, \boldsymbol{z} +\!\!+ \boldsymbol{z}') \coloneqq \llbracket M \rrbracket^t(\boldsymbol{\theta}, (\xi^{(1)}(\boldsymbol{\theta}, \boldsymbol{z}), \dots, \xi^{(\ell)}(\boldsymbol{\theta}, \boldsymbol{z})), \boldsymbol{z}')$$

Proof. The claim is proven by induction on the typing judgements. We focus on the most interesting cases:

– Suppose $\boldsymbol{\theta}, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid [\mathcal{D}] \vdash_{\mathrm{unif}} \mathbf{sample}_{\mathcal{D}} \rhd \varphi_{\boldsymbol{\theta}} : R^{(\mathbf{t},\{\alpha_j\})}$. Let $n \in \mathbb{N}$ and $\xi_1 \in \mathcal{DGS}_{\tau_1}^{(n,\mathrm{dict})}, \dots, \xi_\ell \in \mathcal{DGS}_{\tau_\ell}^{(n,\mathrm{dict})}$. Clearly,

$$(\llbracket \mathbf{sample}_{\mathcal{D}} \rhd \varphi_{\boldsymbol{\theta}} \rrbracket^t * \langle \xi_1, \dots, \xi_\ell \rangle)(\boldsymbol{\theta}, [z_1, \dots, z_{n+1}]) = z_{n+1}$$

Therefore,

$$\llbracket \mathbf{sample}_{\mathcal{D}} \rhd \varphi_{\boldsymbol{\theta}} \rrbracket^t * \langle \xi_1, \dots, \xi_\ell \rangle \in \mathcal{DGS}_{R^{(\mathbf{t},\{\alpha_j\})}}^{(n+1,\ \mathrm{dict}[n+1 \mapsto \alpha_j])}$$

– Suppose

$$\boldsymbol{\theta}, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma_1 +\!\!+ \cdots +\!\!+ \Sigma_\ell \vdash_{\text{unif}} \underline{f}(M_1, \dots, M_k) : \iota^{(\mathbf{t}, \text{dep}_1 \cup \cdots \cup \text{dep}_k)}$$

because $\nabla f \neq \mathbf{0}$ a.e., the dep_i are disjoint and

$$\boldsymbol{\theta}, x_1 : \tau_1, \dots, x_\ell : \tau_\ell \mid \Sigma_i \vdash_{\text{unif}} M_i : \iota_i^{(\mathbf{t}, \text{dep}_i)}$$

Let $n \in \mathbb{N}$ and $\xi_1 \in \mathcal{DGS}_{\tau_1}^{(n,\text{dict})}, \dots, \xi_\ell \in \mathcal{DGS}_{\tau_\ell}^{(n,\text{dict})}$. By the inductive hypothesis,

$$[\![M_i]\!]^t * \langle \xi_1, \dots, \xi_\ell \rangle \in \mathcal{DGS}_{\iota_i}^{(n+|\Sigma_i|, \text{dict}_i)}$$

for some dict_i extending dict. As a consequence, for $1 \leq i \leq k$ there exist partitions $U_{i,1}, \dots, U_{i,N}$ of $\mathbb{R}^{n+|\Sigma_i|}$ and $f_{i,1}, \dots, f_{i,N} : \mathbb{R}^{|\text{dict}_i^{-1}(\text{dep}_i)|} \to \mathbb{R}$ such that for $\boldsymbol{z} +\!\!+ \boldsymbol{z}_i \in U_{i,j}$,

$$\left([\![M_i]\!]^t * \langle \xi_1, \dots, \xi_\ell \rangle\right)(\boldsymbol{\theta}, \boldsymbol{z} +\!\!+ \boldsymbol{z}_i) = f_{i,j}(\boldsymbol{\pi}_{\text{dict}_i^{-1}(\text{dep}_i)}(\boldsymbol{z} +\!\!+ \boldsymbol{z}_i))$$

and $\nabla f_{i,j} \neq 0$ a.e. Now, we define

$$U_{j_1,\dots,j_k} \coloneqq \{\boldsymbol{z} +\!\!+ \boldsymbol{z}_1 +\!\!+ \cdots +\!\!+ \boldsymbol{z}_k \mid \boldsymbol{z} +\!\!+ \boldsymbol{z}_1 \in U_{j_1}, \dots, \boldsymbol{z} +\!\!+ \boldsymbol{z}_k \in U_{j_k}\}$$

$$\text{dict}'(j) \coloneqq \begin{cases} \text{dict}(i) & \text{if } 1 \leq i \leq n \\ \text{dict}_i(j) & \text{if } n + n_1 + \cdots + n_{i-1} + 1 \leq j \leq n + n_1 + \cdots + n_i \end{cases}$$

We can furthermore define $f_{j_1,\dots,j_k} : \mathbb{R}^{|\text{dict}^{-1}(\text{dep}_1 \cup \cdots \cup \text{dep}_\ell)|} \to \mathbb{R}$ satisfying

$$\begin{aligned} f_{j_1,\dots,j_k}(\boldsymbol{\pi}_{\text{dict}^{-1}(\text{dep}_1 \cup \cdots \cup \text{dep}_\ell)}(\boldsymbol{z} +\!\!+ \boldsymbol{z}_1 +\!\!+ \cdots \boldsymbol{z}_k)) = \\ f(f_{j_1}(\boldsymbol{\pi}_{\text{dict}^{-1}(\text{dep}_1)}(\boldsymbol{z} +\!\!+ \boldsymbol{z}_1)), \dots, f_{j_k}(\boldsymbol{\pi}_{\text{dict}^{-1}(\text{dep}_k)}(\boldsymbol{z} +\!\!+ \boldsymbol{z}_k))) \end{aligned} \tag{B.2}$$

for $\boldsymbol{z} +\!\!+ \boldsymbol{z}_1 +\!\!+ \cdots \boldsymbol{z}_k \in U_{j_1,\dots,j_k}$. In particular,

$$\begin{aligned} \left([\![\underline{f}(M_1, \dots, M_k)]\!]^t * \langle \xi_1, \dots, \xi_\ell \rangle\right)(\boldsymbol{\theta}, \boldsymbol{z} +\!\!+ \boldsymbol{z}_1 +\!\!+ \cdots +\!\!+ \boldsymbol{z}_k) \\ = f_{j_1,\dots,j_k}(\boldsymbol{\pi}_{\text{dict}^{-1}(\text{dep}_1 \cup \cdots \cup \text{dep}_\ell)}(\boldsymbol{z} +\!\!+ \boldsymbol{z}_1 +\!\!+ \cdots \boldsymbol{z}_k)) \end{aligned}$$

Note that by the assumption that $\text{dep}_i \cap \text{dep}_{i'} = \emptyset$ and dict_i and $\text{dict}_{i'}$ extend $\text{dict} : \{1, \dots, n\} \to \text{Dep}$,

$$\text{dict}_i^{-1}(\text{dep}_i) \cap \text{dict}_{i'}^{-1}(\text{dep}_{i'}) \cap \{1, \dots, \underbrace{|\boldsymbol{z}|}_{n}\} = \emptyset$$

Consequently, crucially, Item (ii)b follows from Eq. (B.2) and Lemma 2.32 (after renaming variables).

– For conditionals the partitions for the branches can be refined with the guard in the expected manner. □

Lemma 5.38. *Let $f : U_1 \times \cdots \times U_\ell \to \mathbb{R}$ (for open and connected $U_1, \dots, U_\ell \subseteq \mathbb{R}$) and $g_\eta^{(i)}, g^{(i)} : \boldsymbol{\Theta} \times \mathbb{R}^n \to U_i$ for $1 \leq i \leq \ell$ and $\eta > 0$. Suppose the following holds*

(i) *f is continuously differentiable*

(ii) $g^{(i)}(U)$ *is bounded for* $1 \le i \le n$ *and all bounded* $U \subseteq \boldsymbol{\Theta} \times \mathbb{R}^n$
(iii) $g_\eta^{(i)} \xrightarrow{\text{u.a.u.}} g^{(i)}$ *for* $1 \le i \le n$

Then $f \circ \langle g_\eta, h_\eta \rangle \xrightarrow{\text{u.a.u.}} f \circ \langle g, h \rangle : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$.

Proof. Suppose $U \subseteq \mathbb{R}^n$ is bounded and $\epsilon > 0$.

By uniform almost uniform convergence there exist $U^{(i)} \subseteq U$ such that $\text{Leb}(U^{(i)}) < \frac{\epsilon}{\ell}$ and $g_\eta^{(i)} \xrightarrow{\text{unif.}} g^{(i)}$ as $\eta \searrow 0$ on $\boldsymbol{\Theta} \times (U \setminus U^{(i)})$. Define $U' \coloneqq \bigcup_{i=1}^{\ell} U^{(i)}$. Clearly, $\text{Leb}(U') < \epsilon$ and we claim $f_\eta \xrightarrow{\text{unif.}} f$ as $\eta \searrow 0$ on $\boldsymbol{\Theta} \times (U \setminus U')$.

To see this, let $\epsilon' > 0$. Since $\boldsymbol{\Theta}$ is bounded (Assumption 5.8) and by the first two premises,

$$d_k \coloneqq \sup \left\{ \left\| \nabla f(g^{(1)}(\boldsymbol{\theta}, \boldsymbol{s}) + \delta_1, \ldots, g^{(\ell)}(\boldsymbol{\theta}, \boldsymbol{s}) + \delta_\ell) \right\| \mid (\boldsymbol{\theta}, \boldsymbol{z}) \in \boldsymbol{\Theta} \times U, -1 < \delta_1, \ldots, \delta_\ell < 1 \right\}$$

is well-defined.

By uniform convergence of $g_\eta^{(i)} \xrightarrow{\text{unif.}} g^{(i)}$ on $\mathbb{R}^n \setminus U'$ there exists $\eta^{(i)} > 0$ satisfying

$$|g_\eta^{(i)}(\boldsymbol{\theta}, \boldsymbol{z}) - g^{(i)}(\boldsymbol{\theta}, \boldsymbol{z})| \le \frac{\epsilon'}{\sqrt{\ell} \cdot d}$$

for all $0 < \eta < \eta^{(i)}$ and $(\boldsymbol{\theta}, \boldsymbol{z}) \in \boldsymbol{\Theta} \times (U \setminus U')$. Consequently, by the mean value theorem and the Cauchy-Schwarz inequality, for all $0 < \eta < \min_{i=1}^{\ell} \eta^{(i)}$ and $(\boldsymbol{\theta}, \boldsymbol{z}) \in \boldsymbol{\Theta} \times (U \setminus U')$,

$$\begin{aligned}
&\left| f(g_\eta^{(1)}(\boldsymbol{\theta}, \boldsymbol{s}), \ldots, g_\eta^{(\ell)}(\boldsymbol{\theta}, \boldsymbol{s})) - f(g^{(1)}(\boldsymbol{\theta}, \boldsymbol{s}), \ldots, g^{(\ell)}(\boldsymbol{\theta}, \boldsymbol{s})) \right| \\
&\le d \cdot \left\| \left(g_\eta^{(1)}(\boldsymbol{\theta}, \boldsymbol{s}) - g^{(1)}(\boldsymbol{\theta}, \boldsymbol{s}), \ldots, g_\eta^{(\ell)}(\boldsymbol{\theta}, \boldsymbol{s}) - g^{(\ell)}(\boldsymbol{\theta}, \boldsymbol{s}) \right) \right\| \\
&< d \cdot \sqrt{\ell} \cdot \frac{\epsilon'}{\sqrt{\ell} \cdot d} \\
&\le \epsilon'
\end{aligned}$$

□

Appendix C

Supplementary Materials for Chapter 6

C.1 Supplementary Materials for Section 6.2

Lemma 6.3. *If $f : U \to \mathbb{R}$ is a Schwartz function, $\boldsymbol{\varphi}_{(-)} : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}^n$ is a polynomial and a strong diffeomorphism, and $p : \mathbb{R}^n \to \mathbb{R}$ is a polynomial then*

$$\int_U \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left| \frac{\partial f_{(-)}}{\partial \theta_i}(\boldsymbol{\theta}, \boldsymbol{z}) \cdot p(\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z} < \infty$$
$$\int_U \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left| \frac{\partial f_{(-)}}{\partial z_i}(\boldsymbol{\theta}, \boldsymbol{z}) \cdot p(\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z} < \infty$$
$$\int_U \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left| \frac{\partial^2 f_{(-)}}{\partial \theta_i \partial \theta_j}(\boldsymbol{\theta}, \boldsymbol{z}) \cdot p(\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z} < \infty$$

where $f_{\boldsymbol{\theta}}(\boldsymbol{z}) \coloneqq f(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) \cdot |\det \mathbf{J}\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})|$.

Proof. As for Lemma 5.24, by Lemma 2.24 it suffices to prove

$$\sup_{(\boldsymbol{\theta},\boldsymbol{z})\in\boldsymbol{\Theta}\times U} \|\boldsymbol{z}\|^{n+3} \cdot \left| \frac{\partial f_{(-)}}{\partial \theta_i} \cdot p(\boldsymbol{z}) \right|$$

for the first claim. Recall that

$$|\det \mathbf{J}\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})| = \frac{1}{|\det \mathbf{J}\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))|} = \frac{1}{p_1(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))}$$

for a polynomial $p_1 : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}$, which we can assume by Definition 4.2(ii) w.l.o.g. to be positive and greater than a positive constant c. Besides,

$$0 = \mathbf{J}_{\theta_i}(\boldsymbol{\varphi}_{\boldsymbol{\theta}} \circ \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1})(\boldsymbol{z}) = \mathbf{J}_{\theta_i}\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) + \mathbf{J}_{\boldsymbol{s}}\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) \cdot \mathbf{J}_{\theta_i}\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})$$

and hence,

$$\begin{aligned}\mathbf{J}_{\theta_i}\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}) &= -\left(\mathbf{J}_{\boldsymbol{s}}\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))\right)^{-1} \cdot \mathbf{J}_{\theta_i}\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) \\ &= -\frac{1}{\det \mathbf{J}\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))} \cdot \operatorname{adj}(\mathbf{J}\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))) \cdot \mathbf{J}_{\theta_i}\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))\end{aligned}$$

$$= -\frac{1}{p(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))} \cdot \boldsymbol{p_2}(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))$$

for a suitable[1] polynomial $\boldsymbol{p_2} : \boldsymbol{\Theta} \times \mathbb{R}^n \to \mathbb{R}^n$. Therefore,

$$\begin{aligned}\frac{\partial (f \circ \boldsymbol{\varphi}_{(-)}^{-1})}{\partial \theta_i}(\boldsymbol{z}) &= (\nabla f)(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) \cdot \mathbf{J}_{\theta_i} \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}) \\ &= \frac{(\nabla f)(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) \cdot \boldsymbol{p_2}(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))}{p(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))}\end{aligned}$$

and consequently,

$$\begin{aligned}&\sup_{(\boldsymbol{\theta},\boldsymbol{z}) \in \boldsymbol{\Theta} \times U} \left| \|\boldsymbol{z}\|^{n+3} \cdot \frac{\partial (f \circ \boldsymbol{\varphi}_{(-)}^{-1})}{\partial \theta_i}(\boldsymbol{z}) \cdot \det \mathbf{J} \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}) \cdot p(\boldsymbol{z}) \right| \\ &\leq \frac{1}{c^2} \cdot \sup_{(\boldsymbol{\theta},\boldsymbol{s}) \in \boldsymbol{\Theta} \times U} \left| \|\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s})\|^{n+3} \cdot (\nabla f(\boldsymbol{s}) \cdot \boldsymbol{p_2}(\boldsymbol{\theta}, \boldsymbol{s})) \cdot p(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \right| < \infty\end{aligned}$$

because the polynomials are uniformly bounded by polynomials independent of $\boldsymbol{\theta}$ (by Lemma 2.26) and derivatives of Schwartz functions are Schwartz functions, too.

Likewise, we can exploit

$$\begin{aligned}\frac{\partial}{\partial \theta_i} \det \mathbf{J} \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}) &= \frac{\partial}{\partial \theta_i} \frac{1}{p_1(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))} \\ &= -\frac{\frac{\partial p_1}{\partial \theta_i}(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) + \nabla_{\boldsymbol{s}} p_1(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) \cdot \mathbf{J}_{\theta_i} \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})}{\left(p_1(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))\right)^2} \\ &= -\frac{\frac{\partial p}{\partial \theta_i}(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))}{\left(p_1(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))\right)^2} - \frac{\nabla_{\boldsymbol{s}} p_1(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) \cdot \boldsymbol{p_2}(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))}{\left(p_1(\boldsymbol{\theta}, \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}))\right)^3}\end{aligned}$$

to show

$$\sup_{(\boldsymbol{\theta},\boldsymbol{z}) \in \boldsymbol{\Theta} \times U} \left| \|\boldsymbol{z}\|^{n+3} \cdot f(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) \cdot \frac{\partial}{\partial \theta_i} \det \mathbf{J} \boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}) \cdot p(\boldsymbol{z}) \right| < \infty$$

The same insights can be used to show the second and third bounds. □

Proposition 6.5. *If* $F \in \Lambda_{\mathrm{s}}^{\mathrm{SFC}}$ *then*

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}}[[\![F]\!]_\eta(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))] \xrightarrow{\text{unif.}} \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\mathcal{D}}}[[\![F]\!](\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))] \qquad \text{as } \eta \searrow 0 \text{ for } \boldsymbol{\theta} \in \boldsymbol{\Theta}$$

Proof. Let $p : \mathbb{R}^n \to \mathbb{R}$ be a polynomial bound to $[\![F]\!]_\eta$ and $[\![F]\!]$ (using Remark 5.13) and let $\epsilon > 0$. We define

$$\begin{aligned}U &:= \operatorname{supp}(\boldsymbol{\mathcal{D}}) \\ c &:= \int_U \left| \sup_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \frac{\partial \boldsymbol{\mathcal{D}}_{(-)}}{\partial \theta_i}(\boldsymbol{\theta}, \boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z} \\ V_k &:= \left\{ \boldsymbol{z} \in \operatorname{supp}(\boldsymbol{\mathcal{D}}) \mid |f_\eta(\boldsymbol{z}) - f(\boldsymbol{z})| > \frac{\epsilon}{2c} \text{ for some } 0 < \eta < \frac{1}{k} \right\}\end{aligned}$$

[1] adj is the adjugate matrix

$$\mu(V) \coloneqq \int_V p(\boldsymbol{z}) \cdot \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left|\frac{\partial \boldsymbol{\mathcal{D}}_{(-)}}{\partial \theta_i}(\boldsymbol{\theta}, \boldsymbol{z})\right| \mathrm{d}\boldsymbol{z}$$

which is a finite measure by Lemma 6.3. Note that $(V_k)_{k\in\mathbb{N}}$ is a non-increasing sequence of sets and $\bigcap_{k\in\mathbb{N}} V_k$ is negligible. Hence, by continuity from above (of μ) there exists k such that $\mu(V_k) < \frac{\epsilon}{4}$. Finally, it suffices to observe that for $0 < \eta < \frac{1}{k}$ and $\boldsymbol{\theta} \in \boldsymbol{\Theta}$:

$$\begin{aligned}
&\left|\frac{\partial}{\partial\theta_i}\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[[\![F]\!]_\eta(\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))] - \frac{\partial}{\partial\theta_i}\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{D}}}[[\![F]\!](\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\boldsymbol{s}))]\right| \\
&= \left|\frac{\partial}{\partial\theta_i}\mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}[[\![F]\!]_\eta(\boldsymbol{z})] - \frac{\partial}{\partial\theta_i}\mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}}[[\![F]\!](\boldsymbol{z})]\right| \\
&\leq \int_U \left|\frac{\partial \boldsymbol{\mathcal{D}}_{(-)}}{\partial\theta_i}(\boldsymbol{\theta},\boldsymbol{z})\right| \cdot |[\![F]\!]_\eta(\boldsymbol{z}) - [\![F]\!](\boldsymbol{z})| \,\mathrm{d}\boldsymbol{z} \\
&\leq \int_{V_k} \left|\frac{\partial \boldsymbol{\mathcal{D}}_{(-)}}{\partial\theta_i}(\boldsymbol{\theta},\boldsymbol{z})\right| \cdot |[\![F]\!]_\eta(\boldsymbol{z}) - [\![F]\!](\boldsymbol{z})| \,\mathrm{d}\boldsymbol{z} \\
&\qquad + \int_{U\setminus V_k} \left|\frac{\partial \boldsymbol{\mathcal{D}}_{(-)}}{\partial\theta_i}(\boldsymbol{\theta},\boldsymbol{z})\right| \cdot |[\![F]\!]_\eta(\boldsymbol{z}) - [\![F]\!](\boldsymbol{z})| \,\mathrm{d}\boldsymbol{z} \\
&\leq 2 \cdot \int_{V_k} \left|\frac{\partial \boldsymbol{\mathcal{D}}_{(-)}}{\partial\theta_i}(\boldsymbol{\theta},\boldsymbol{z})\right| \cdot p(\boldsymbol{z}) \,\mathrm{d}\boldsymbol{z} + \int_{U\setminus V_k} \left|\frac{\partial \boldsymbol{\mathcal{D}}_{(-)}}{\partial\theta_i}(\boldsymbol{\theta},\boldsymbol{z})\right| \cdot \frac{\epsilon}{2c} \,\mathrm{d}\boldsymbol{z} \\
&\leq 2 \cdot \mu(V_k) + \frac{\epsilon}{2c} \cdot \int_U \left|\frac{\partial \boldsymbol{\mathcal{D}}_{(-)}}{\partial\theta_i}(\boldsymbol{\theta},\boldsymbol{z})\right| \mathrm{d}\boldsymbol{z} \\
&\leq \epsilon
\end{aligned}$$

□

C.1.1 Supplementary Materials for Section 6.2.1

Lemma 6.10. *Let $F \in \Lambda_\ell^{\mathrm{SFC}}$, $f : U \to \mathbb{R}$ be a non-negative Schwartz function, where $U = U_1 \times \cdots \times U_n$ and $U_1, \ldots, U_n \in \{\mathbb{R}, \mathbb{R}_{\geq 0}\}$, and p be a non-negative polynomial. For all $1 \leq i \leq n$ there exists $c > 0$ such that for all $0 < \eta \leq 1$,*

$$\int_U f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p(\boldsymbol{z}) \cdot \left|\frac{\partial^2 [\![F]\!]_\eta}{\partial z_i^2}(\boldsymbol{z})\right| \mathrm{d}\boldsymbol{z} < c \cdot \eta^{-\ell}$$

$$\int_U f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p(\boldsymbol{z}) \cdot \left|\frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z})\right|^2 \mathrm{d}\boldsymbol{z} < c \cdot \eta^{-\ell}$$

where $f_{\boldsymbol{\theta}} \coloneqq f(\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})) \cdot |\det \mathbf{J}\boldsymbol{\varphi}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z})|$.

Proof. Note that $f_{\boldsymbol{\theta}}$ is differentiable and non-negative. To simplify notation, we assume that $U = \mathbb{R}^n$. (Otherwise the proof is similar, exploiting Lemma 6.6.) Besides, it suffices to establish the first bound. To see that the second is a consequence of the first, we use integration by parts

$$\begin{aligned}
&\int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p(\boldsymbol{z}) \cdot \left(\frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z})\right)^2 \mathrm{d}\boldsymbol{z} \\
&= \int\int \left(f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p(\boldsymbol{z}) \cdot \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z})\right) \cdot \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z}) \,\mathrm{d}z_j \,\mathrm{d}\boldsymbol{z}_{-j} \\
&= \int \underbrace{\left[f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p(\boldsymbol{z}) \cdot \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z}) \cdot [\![F]\!]_\eta(\boldsymbol{z})\right]_{-\infty}^{\infty}}_{=0} \mathrm{d}\boldsymbol{z}_{-j}
\end{aligned}$$

$$-\int \frac{\partial}{\partial z_i}\left(f_{\boldsymbol{\theta}}(\boldsymbol{z})\cdot p(\boldsymbol{z})\cdot \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z})\right)\cdot [\![F]\!]_\eta(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z}$$

where $\boldsymbol{z}_{-j}$ is $z_1,\dots,z_{j-1},z_{j+1},\dots,j_n$, because by Lemma 5.24, for fixed $\eta>0$ and $\boldsymbol{z}_{-j}$,

$$\lim_{z_j\to\pm\infty} f_{\boldsymbol{\theta}}(\boldsymbol{z})\cdot \underbrace{p(\boldsymbol{z})\cdot \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z})\cdot [\![F]\!]_\eta(\boldsymbol{z})}_{\text{bounded by polynomial}}$$

We continue bounding:

$$\begin{aligned}
&-\int \frac{\partial}{\partial z_i}\left(f_{\boldsymbol{\theta}}(\boldsymbol{z})\cdot p(\boldsymbol{z})\cdot \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z})\right)\cdot [\![F]\!]_\eta(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z}\\
&=-\int \frac{\partial f_{(-)}}{\partial z_i}(\boldsymbol{\theta},\boldsymbol{z})\cdot p(\boldsymbol{z})\cdot \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z})\cdot [\![F]\!]_\eta(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z}\\
&\quad -\int f_{\boldsymbol{\theta}}(\boldsymbol{z})\cdot \frac{\partial p}{\partial z_i}(\boldsymbol{z})\cdot \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z})\cdot [\![F]\!]_\eta(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z}\\
&\quad -\int f_{\boldsymbol{\theta}}(\boldsymbol{z})\cdot p(\boldsymbol{z})\cdot \frac{\partial^2 [\![F]\!]_\eta}{\partial z_i^2}(\boldsymbol{z})\cdot [\![F]\!]_\eta(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z}\\
&\le \eta^{-\ell}\cdot\int \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\left|\frac{\partial f_{(-)}}{\partial z_i}(\boldsymbol{\theta},\boldsymbol{z})\cdot p_1(\boldsymbol{z})\right|\mathrm{d}\boldsymbol{z}\\
&\quad +\eta^{-\ell}\cdot\int \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\left|f_{\boldsymbol{\theta}}(\boldsymbol{z})\cdot p_2(\boldsymbol{z})\right|\mathrm{d}\boldsymbol{z}\\
&\quad +\int f_{\boldsymbol{\theta}}(\boldsymbol{z})\cdot p_3(\boldsymbol{z})\cdot\left|\frac{\partial^2 [\![F]\!]_\eta}{\partial z_i^2}(\boldsymbol{z})\right|
\end{aligned}$$

for suitable polynomial bounds $p_1,p_2,p_3:\mathbb{R}^n\to\mathbb{R}$ (which exist due to the fact that $\frac{\partial p}{\partial z_i}$ *is* a polynomial, Lemmas 2.26(i) and 6.6 and Remark 5.13) and the second inequality follows with Lemma 6.3.

We prove the claim by induction on the definition of $\Lambda_\ell^{\mathrm{SFC}}$:

- For $z_j\in\Lambda_0^{\mathrm{SFC}}$ and $F\in\Lambda_{\ell+1}^{\mathrm{SFC}}$ due to $F\in\Lambda_\ell^{\mathrm{SFC}}$ the claim is obvious.
- For $g(F_1,\dots,F_k)\in\Lambda_\ell^{\mathrm{SFC}}$ because $F_1,\dots,F_k\in\Lambda_\ell^{\mathrm{SFC}}$,

$$\begin{aligned}
&\int f_{\boldsymbol{\theta}}(\boldsymbol{z})\cdot p(\boldsymbol{z})\cdot\left|\frac{\partial^2(g\circ\langle [\![F_1]\!]_\eta,\dots,[\![F_k]\!]_\eta\rangle)}{\partial z_i^2}(\boldsymbol{z})\right|\mathrm{d}\boldsymbol{z}\\
&\le \sum_{1\le j,j'\le k}\int f_{\boldsymbol{\theta}}(\boldsymbol{z})\cdot p(\boldsymbol{z})\cdot\left|\frac{\partial^2 g}{\partial y_j\partial y_{j'}}([\![F_1]\!]_\eta(\boldsymbol{z}),\dots,[\![F_k]\!]_\eta(\boldsymbol{z}))\cdot\frac{\partial [\![F_j]\!]_\eta}{\partial z_i}(\boldsymbol{z})\cdot\frac{\partial [\![F_{j'}]\!]_\eta}{\partial z_i}(\boldsymbol{z})\right|\mathrm{d}\boldsymbol{z}\\
&\quad +\sum_{1\le j\le k}\int f_{\boldsymbol{\theta}}(\boldsymbol{z})\cdot p(\boldsymbol{z})\cdot\left|\frac{\partial g}{\partial y_j}([\![F_1]\!]_\eta(\boldsymbol{z}),\dots,[\![F_k]\!]_\eta(\boldsymbol{z}))\cdot\frac{\partial^2 [\![F_j]\!]_\eta}{\partial^2 z_i}(\boldsymbol{z})\right|\mathrm{d}\boldsymbol{z}
\end{aligned}$$

  By Assumption 6.2, $\frac{\partial g}{\partial y_j}(g_\eta^{(1)}(\boldsymbol{z}),\dots,g_\eta^{(\ell)}(\boldsymbol{z}))$ is bounded by a polynomial. Therefore the second summand can be bounded by the inductive hypothesis. For the first, again by Assumption 6.2, we can bound

$$p(\boldsymbol{z})\cdot\left|\frac{\partial^2 g}{\partial y_j\partial y_{j'}}(g_\eta^{(1)}(\boldsymbol{z}),\dots,g_\eta^{(\ell)}(\boldsymbol{z}))\right|\le p_1(\boldsymbol{z})$$

for a (non-negative) polynomial p_1, apply the Cauchy-Schwarz inequality and apply the inductive hypothesis to

$$\int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p_1(\boldsymbol{z}) \cdot \left| \frac{\partial [\![F_j]\!]_\eta}{\partial z_i}(\boldsymbol{z}) \right|^2 \mathrm{d}\boldsymbol{z} \qquad \int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p_1(\boldsymbol{z}) \cdot \left| \frac{\partial [\![F_{j'}]\!]_\eta}{\partial z_i}(\boldsymbol{z}) \right|^2 \mathrm{d}\boldsymbol{z}$$

– Next, suppose **if** $F_1 < 0$ **then** F_2 **else** $F_3 \in \Lambda_{\ell+1}^{\mathrm{SFC}}$ because $F_1 \in \Lambda_\ell^{\mathrm{SFC}}$ and $F_2, F_3 \in \Lambda_{\ell+1}^{\mathrm{SFC}}$. By linearity we bound (similarly for the other branch):

$$\begin{aligned}
&\int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p(\boldsymbol{z}) \cdot \left| \frac{\partial^2((\sigma_\eta \circ [\![F_1]\!]_\eta) \cdot [\![F_3]\!]_\eta)}{\partial z_i^2}(\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z} \\
&\leq \int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p(\boldsymbol{z}) \cdot \left| \frac{\partial^2(\sigma_\eta \circ [\![F_1]\!]_\eta)}{\partial z_i^2}(\boldsymbol{z}) \cdot [\![F_3]\!]_\eta(\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z} \\
&\quad + 2 \cdot \int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p(\boldsymbol{z}) \cdot \left| \frac{\partial(\sigma_\eta \circ [\![F_1]\!]_\eta)}{\partial z_i}(\boldsymbol{z}) \cdot \frac{\partial [\![F_3]\!]_\eta}{\partial z_i}(\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z} \\
&\quad + \int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p(\boldsymbol{z}) \cdot \left| \sigma_\eta([\![F_1]\!]_\eta(\boldsymbol{z})) \cdot \frac{\partial^2 [\![F_3]\!]_\eta}{\partial z_i^2}(\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z}
\end{aligned}$$

We abbreviate $F \equiv F_1$ and bound $p(\boldsymbol{z}) \cdot |[\![F_3]\!]_\eta(\boldsymbol{z})|$ by the non-negative polynomial p_2 (using Remark 5.13). Bounding the first summand is most interesting:

$$\begin{aligned}
&\int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p(\boldsymbol{z}) \cdot \left| \frac{\partial^2(\sigma_\eta \circ [\![F_1]\!]_\eta)}{\partial z_i^2}(\boldsymbol{z}) \cdot [\![F_3]\!]_\eta(\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z} \\
&\leq \int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p_2(\boldsymbol{z}) \cdot \left| \frac{\partial^2(\sigma_\eta \circ [\![F]\!]_\eta)}{\partial z_i^2}(\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z} \\
&\leq \int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p_2(\boldsymbol{z}) \cdot \left| \sigma_\eta''([\![F]\!]_\eta(\boldsymbol{z})) \right| \cdot \left(\frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z}) \right)^2 \mathrm{d}\boldsymbol{z} \\
&\quad + \int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p_2(\boldsymbol{z}) \cdot \left| \underbrace{\sigma_\eta'([\![F]\!]_\eta(\boldsymbol{z}))}_{\leq \eta^{-1}} \cdot \frac{\partial^2 [\![F]\!]_\eta}{\partial z_i^2}(\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z}
\end{aligned}$$

The second summand can be bounded with the inductive hypothesis. For the first summand we exploit that

$$\begin{aligned}
|\sigma_\eta''(x)| &= \eta^{-2} \left| \sigma_\eta(x) \cdot (1 - \sigma_\eta(x)) \cdot (1 - 2\sigma_\eta(x)) \right| \\
&\leq \eta^{-2} \cdot \sigma_\eta(x) \cdot (1 - \sigma_\eta(x)) \\
&= \eta^{-1} \cdot \sigma_\eta'(x)
\end{aligned}$$

and continue using integration by parts again in the second step

$$\begin{aligned}
&\int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p_2(\boldsymbol{z}) \cdot \left| \sigma_\eta''([\![F]\!]_\eta(\boldsymbol{z})) \cdot \left(\frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z}) \right)^2 \right| \mathrm{d}\boldsymbol{z} \\
&\leq \eta^{-1} \cdot \int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p_2(\boldsymbol{z}) \cdot \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z}) \cdot \frac{\partial(\sigma_\eta \circ [\![F]\!]_\eta)}{\partial z_i}(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} \\
&= -\eta^{-1} \cdot \int \frac{\partial}{\partial z_i} \left(f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p_2(\boldsymbol{z}) \cdot \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z}) \right) \cdot \sigma_\eta([\![F]\!]_\eta(\boldsymbol{z})) \, \mathrm{d}\boldsymbol{z}
\end{aligned}$$

$$
\begin{aligned}
&= -\eta^{-1} \cdot \int \frac{\partial f_{(-)}}{\partial z_i}(\boldsymbol{\theta}, \boldsymbol{z}) \cdot p_2(\boldsymbol{z}) \cdot \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z}) \cdot \sigma_\eta([\![F]\!]_\eta(\boldsymbol{z}))\, \mathrm{d}\boldsymbol{z} \\
&\quad - \eta^{-1} \cdot \int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot \frac{\partial p_2}{\partial z_i}(\boldsymbol{z}) \cdot \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z}) \cdot \sigma_\eta([\![F]\!]_\eta(\boldsymbol{z}))\, \mathrm{d}\boldsymbol{z} \\
&\quad - \eta^{-1} \cdot \int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot p_2(\boldsymbol{z}) \cdot \frac{\partial^2 [\![F]\!]_\eta}{\partial z_i^2}(\boldsymbol{z}) \cdot \sigma_\eta([\![F]\!]_\eta(\boldsymbol{z}))\, \mathrm{d}\boldsymbol{z} \\
&\leq \eta^{-1} \cdot \int \left| \frac{\partial f_{(-)}}{\partial z_i}(\boldsymbol{\theta}, \boldsymbol{z}) \cdot p_2(\boldsymbol{z}) \cdot \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z} \\
&\quad + \eta^{-1} \cdot \int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot \left| \frac{\partial p_2}{\partial z_i}(\boldsymbol{z}) \cdot \frac{\partial [\![F]\!]_\eta}{\partial z_i}(\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z} \\
&\quad + \eta^{-1} \cdot \int f_{\boldsymbol{\theta}}(\boldsymbol{z}) \cdot \left| p_2(\boldsymbol{z}) \cdot \frac{\partial^2 [\![F]\!]_\eta}{\partial z_i^2}(\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z}
\end{aligned}
$$

and the claim follows with the inductive hypothesis (recall $F \equiv F_1 \in \Lambda_\ell^{\mathrm{SFC}}$), Lemmas 6.3 and 6.6. □

Appendix D

Supplementary Materials for Chapter 8

D.1 Supplementary Materials for Section 8.1

Lemma D.1. *Let $\mathcal{D}$ be a Schwartz distribution and*

$$f(\boldsymbol{\theta}, \boldsymbol{s}) = \sum_{i=1}^{\ell} [(\boldsymbol{\theta}, \boldsymbol{s}) \in U_i] \cdot f_i(\boldsymbol{\theta}, \boldsymbol{s})$$

be continuous, where $U_1, \ldots, U_\ell$ is a partition of $\boldsymbol{\Theta} \times \mathbb{R}^n$ and the partial derivatives of $f_1, \ldots, f_\ell$ are bounded by polynomials. Then the third premise of Lemma 8.3 is satisfied: for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ there exists an integrable $g : \mathbb{R}^n \to \mathbb{R}$ and $\delta > 0$ satisfying

$$\left| \frac{f(\boldsymbol{\theta} + h \cdot \mathbf{e}_i, \boldsymbol{s}) - f(\boldsymbol{\theta}, \boldsymbol{s})}{h} \right| \leq g(\boldsymbol{s})$$

for all $0 < |h| < \delta$ and $\boldsymbol{s} \in \mathbb{R}^n$.

Proof. Again, it suffices to prove the one-dimensional case. Let θ_0 be arbitrary. Suppose p_i is a uniform polynomial bound on $\frac{\partial f_i}{\partial \theta}$. (NB $\boldsymbol{s} \mapsto \mathcal{D}(\boldsymbol{s}) \cdot p_i(\boldsymbol{s})$ is integrable.) We claim that

$$\left| \frac{f(\theta, \boldsymbol{s}) - f(\theta', \boldsymbol{s})}{\theta - \theta'} \right| \leq \sum_{i=1}^{\ell} p_i(\boldsymbol{s})$$

for all $\theta, \theta' \in B_1(\theta_0)$ and $\boldsymbol{s}$. (Thus, we can choose $\delta := 1$.)

Fix $\boldsymbol{s}$ and θ. Let i be such that $(\theta, \boldsymbol{s}) \in U_i$. For $\theta' \geq \theta$ define

$$J_{\theta'} := \{1 \leq j \leq \ell \mid j \neq i \text{ and } (\theta'', \boldsymbol{s}) \in U_j \text{ for some } \theta \leq \theta'' < \theta'\}$$

We prove

$$\left| \frac{f(\theta, \boldsymbol{s}) - f(\theta', \boldsymbol{s})}{\theta - \theta'} \right| \leq p_i(\boldsymbol{s}) + \sum_{j \in J_{\theta'}}^{\ell} p_j(\boldsymbol{s})$$

for all $\theta \leq \theta' \leq \theta + 1$ by induction on $|J_{\theta'}|$.

If $|J_{\theta'}| = 0$ then $\theta' \leq \sup\{\theta'' \mid \theta \leq \theta'' < \theta' \mid (\theta'', \boldsymbol{s}) \in U_i\}$ and by continuity

$$\left|\frac{f(\theta, \boldsymbol{s}) - f(\theta', \boldsymbol{s})}{\theta - \theta'}\right| = \left|\frac{f_i(\theta, \boldsymbol{s}) - f_i(\theta', \boldsymbol{s}}{\theta - \theta'}\right| \leq p_i(\boldsymbol{s})$$

Otherwise let j be such that $(\theta', \boldsymbol{s}) \in U_j$. Define $\theta_j \coloneqq \inf\{\theta'' \mid \theta \leq \theta'' \leq \theta' \mid (\theta'', \boldsymbol{s}) \in U_j\}$. By continuity, $f(\theta_j, \boldsymbol{s}) = f_j(\theta_j, \boldsymbol{s})$. Note that $j \notin J_{\theta_j}$. Therefore, the inductive is applicable and yields:

$$\left|\frac{f(\theta, \boldsymbol{s}) - f(\theta_j, \boldsymbol{s}}{\theta - \theta_j}\right| \leq p_i(\boldsymbol{s}) + \sum_{k \in J_{\theta_j}}^{\ell} p_k(\boldsymbol{s})$$

Consequently,

$$\begin{aligned}
\left|\frac{f(\theta, \boldsymbol{s}) - \theta_j, \boldsymbol{s}}{\theta - \theta'}\right| &\leq \left|\frac{f(\theta, \boldsymbol{s}) - f(\theta_j, \boldsymbol{s})}{\theta - \theta'}\right| + \left|\frac{f(\theta_j, \boldsymbol{s}) - f(\theta', \boldsymbol{s})}{\theta - \theta'}\right| \\
&\leq \left|\frac{f(\theta, \boldsymbol{s}) - f(\theta_j, \boldsymbol{s})}{\theta - \theta_j}\right| + \left|\frac{f_j(\theta_j, \boldsymbol{s}) - f_j(\theta', \boldsymbol{s})}{\theta_j - \theta'}\right| \\
&\leq p_i(\boldsymbol{s}) + \sum_{k \in J_{\theta_j}} p_k(\boldsymbol{s}) + p_j(\boldsymbol{s})
\end{aligned}$$

□

D.2 Supplementary Materials for Section 8.2

Lemma D.2. *The resolvent of a definite clause $B \to P\, x_1 \cdots x_n\, t'$ and a goal clause*

$$\varphi_1 \wedge \cdots \wedge \varphi_\ell \wedge s_1\, z_1 \wedge \cdots \wedge s_m\, z_m \to \bot$$

where $s_i\, z_i \equiv P\, t_1 \cdots t_n\, z_i$ for $1 \leq i \leq m$, is a goal clause.

Proof. All variables in the resolvent have base sort because by definition of goal clauses, $t_1, \ldots, t_n$ only contain variables of base sort and the only non-base sort variables are amongst $x_1, \ldots, x_n$.

For every foreground atom $s\, z$ in B,

$$(s\, z)[\boldsymbol{t}/\boldsymbol{x}] \equiv s[\boldsymbol{t}/\boldsymbol{x}]\, z$$

because $x_1, \ldots, x_n$ are different from z. Likewise, for every $s_j\, z_j$ $(j \neq i)$ in B',

$$(s_j\, z_j)[t'/z_i][\boldsymbol{t}/\boldsymbol{x}] \equiv (s_j[t'/z_i])[\boldsymbol{t}/\boldsymbol{x}]\, z_j$$

because z_j must be different from z_i and the $x_1, \ldots, x_n$ (the variables of the clauses are assumed to be renamed). Therefore, the variables in trailing positions of the resolvent are also distinct.

Finally, we argue that the second requirement of bodies is satisfied. For $j' < j$, $(s_{j'}[t'/z_i])[\boldsymbol{t}/\boldsymbol{x}]\, z_{j'}$ does not contain z_j because neither t' does so (by the renaming assumption) nor $t_1, \ldots, t_n$ by definition of goal clauses. Besides, for every foreground atom $s\, z$ in B, $s[\boldsymbol{t}/\boldsymbol{x}]\, z$ does not contain z_j if $j > i$: neither $s\, z$ contains z_j (by the renaming assumption) nor $t_1, \ldots, t_n$ (by definition of goal clauses).

Similarly, for every foreground atom $s\, z$ in B, z cannot occur in $(s_j[t'/z_i])[\boldsymbol{t}/\boldsymbol{x}]\, z_j$ provided that $j < i$. □

D.3 Supplementary Materials for Section 8.4

Lemma 8.30 (Fundamental)**.** *Suppose* $x_1 : \tau_1, \ldots, x_m : \tau_m \mid [\mathcal{D}_1, \ldots, \mathcal{D}_n] \vdash_{\text{aff}} M : \tau$ *and* $\mathcal{A} \models \Phi_M$*. Then for all* $f_1 \preceq'_{\tau_1} p_1, \ldots, f_n \preceq'_{\tau_n} p_n$ *and* $z_1, \ldots, z_n \in \mathbb{R}$*,*

$$[\![M]\!]^t(f_1, \ldots, f_m, z_1, \ldots, z_n) \preceq_\tau R_M^{\mathcal{A}}(p_1) \cdots (p_m)(z_1) \cdots (z_n)$$

where $r \preceq'_\iota r'$ *iff* $r = r'$ *and for* $\tau \neq \iota$*,* $f \preceq'_\tau p$ *iff* $f \preceq_\tau p$*.*

Proof. We abbreviate $\Gamma = x_1 : \tau_1, \ldots, x_m : \tau_m$ and prove the claim by structural induction.

– If $\Gamma \mid [] \vdash_{\text{aff}} x_i : \iota^{(a)}$ then

$$\mathrm{Gr}_{x_i}^{\mathcal{A}}(\boldsymbol{f})(\underbrace{f_i}_{[\![x_i]\!]^t(\boldsymbol{f},[])}) = 1$$

immediately due to $\mathcal{A} \models \Phi_{x_i}$. Otherwise $\Gamma \mid [] \vdash_{\text{aff}} x_i : \tau_1 \bullet \Sigma_1 \to \cdots \to \tau_\ell \bullet \Sigma_\ell \to \iota^{(a)}$. To show that

$$[\![x_i]\!]^t(\boldsymbol{f}, []) \preceq_{\tau_1 \bullet \Sigma_1 \to \cdots \to \tau_\ell \bullet \Sigma_\ell \to \iota^{(a)}} \mathrm{Gr}_{x_i}^{\mathcal{A}}(\boldsymbol{p})$$

let $g_1 \preceq'_{\tau_1} q_1, \ldots g_\ell \preceq'_{\tau_\ell} q_\ell$ and $\boldsymbol{z}_1 \in \mathbb{R}^{|\Sigma_1|}, \ldots, \boldsymbol{z}_\ell \in \mathbb{R}^{|\Sigma_\ell|}$. Due to $f_i \preceq_{\tau_i} p_i$,

$$p_i(q_1)(\boldsymbol{z}_1) \cdots (q_\ell)(\boldsymbol{z}_\ell)(f_i(g_1, \boldsymbol{z}_1) \cdots (g_\ell, \boldsymbol{z}_\ell)) = 1$$

and therefore, due to $\mathcal{A} \models \Phi_{x_i}$,

$$\mathrm{Gr}_{x_i}^{\mathcal{A}}(\boldsymbol{p})(q_1)(\boldsymbol{z}_1) \cdots (q_\ell)(\boldsymbol{z}_\ell)(\underbrace{f_i}_{[\![x_i]\!]^t(\boldsymbol{f},[])}(g_1, \boldsymbol{z}_1) \cdots (g_\ell, \boldsymbol{z}_\ell)) = 1$$

– For $\Gamma \mid [\mathcal{D}] \vdash_{\text{aff}} \mathbf{sample}\, \mathcal{D} \rhd \varphi_{\boldsymbol{\theta}}$, due to $\mathcal{A} \models \Phi_M$,

$$\mathrm{Gr}^{\mathcal{A}}_{\mathbf{sample}\, \mathcal{D} \rhd \varphi_{\boldsymbol{\theta}}}(\boldsymbol{p})(z)(\underbrace{z}_{[\![\mathbf{sample}\, \mathcal{D} \rhd \varphi_{\boldsymbol{\theta}}]\!]^t(\boldsymbol{f},z)}) = 1$$

– Suppose $\Gamma \mid \Sigma_1 + \!\!+ \cdots + \!\!+ \Sigma_\ell \vdash_{\text{aff}} \underline{f}(M_1, \ldots, M_\ell) : \iota^{(a)}$ because $\Gamma \mid \Sigma_i \vdash_{\text{aff}} M_i : \iota_i^{(a_i)}$, and let $\boldsymbol{f} \preceq' \boldsymbol{p}$ and $\boldsymbol{z}_1 \in \mathbb{R}^{|\Sigma_1|}, \ldots, \boldsymbol{z}_\ell \in \mathbb{R}^{|\Sigma_\ell|}$. By the inductive hypothesis, $\mathrm{Gr}_{M_i}^{\mathcal{A}}(\boldsymbol{p})(\boldsymbol{z}_i)([\![M_i]\!]^t(\boldsymbol{f}, \boldsymbol{z}_i)) = 1$. Due to $\mathcal{A} \models \Phi_{\underline{f}(M_1,\ldots,M_\ell)}$,

$$\mathrm{Gr}^{\mathcal{A}}_{\underline{f}(M_1,\ldots,M_\ell)}(\boldsymbol{p})(\boldsymbol{z}_1) \cdots (\boldsymbol{z}_\ell)(\underbrace{f([\![M_1]\!]^t(\boldsymbol{f}, \boldsymbol{z}_1), \ldots, [\![M_\ell]\!]^t(\boldsymbol{f}, \boldsymbol{z}_\ell))}_{[\![\underline{f}(M_1,\ldots,M_\ell)]\!]^t(\boldsymbol{f}, \boldsymbol{z}_1 + \!\!+ \cdots \boldsymbol{z}_\ell)}) = 1$$

– Suppose $\Gamma \mid \Sigma_1 + \!\!+ \Sigma_2 + \!\!+ \Sigma_3 \vdash_{\text{aff}} \mathbf{if}\, L < 0\, \mathbf{then}\, M\, \mathbf{else}\, N : \iota^{(a)}$ because $\Gamma \mid \Sigma_1 \vdash_{\text{aff}} L : R^{(\mathbf{t})}$, $\Gamma \mid \Sigma_2 \vdash_{\text{aff}} M : \iota^{(a)}$ and $\Gamma \mid \Sigma_3 \vdash_{\text{aff}} N : \iota^{(a)}$ and $\boldsymbol{f} \preceq' \boldsymbol{p}$.

By the inductive hypothesis,

$$\mathrm{Gr}_L^{\mathcal{A}}(\boldsymbol{p})(\boldsymbol{z}_1)([\![L]\!]^t(\boldsymbol{f}, \boldsymbol{z}_1)) = 1$$
$$\mathrm{Gr}_M^{\mathcal{A}}(\boldsymbol{p})(\boldsymbol{z}_2)([\![M]\!]^t(\boldsymbol{f}, \boldsymbol{z}_2)) = 1$$

$$\mathrm{Gr}^{\mathcal{A}}_{N}(\boldsymbol{p})(\boldsymbol{z}_3)([\![N]\!]^t(\boldsymbol{f},\boldsymbol{z}_3)) = 1$$

If $[\![L]\!]^t(\boldsymbol{f},\boldsymbol{z}_1) < 0$ then by the assumption that $\mathcal{A} \models \Phi_{\mathbf{if}\ L<0\ \mathbf{then}\ M\ \mathbf{else}\ N}$,

$$\mathrm{Gr}^{\mathcal{A}}_{\mathbf{if}\ L<0\ \mathbf{then}\ M\ \mathbf{else}\ N}(\boldsymbol{p})(\boldsymbol{z}_1)(\boldsymbol{z}_2)(\boldsymbol{z}_3)(\underbrace{[\![M]\!]^t(\boldsymbol{f},\boldsymbol{z}_2)}_{[\![\mathbf{if}\ L<0\ \mathbf{then}\ M\ \mathbf{else}\ N]\!]^t(\boldsymbol{f},\boldsymbol{z}_1 +\!\!+ \boldsymbol{z}_2 +\!\!+ \boldsymbol{z}_3)}) = 1$$

otherwise

$$\mathrm{Gr}^{\mathcal{A}}_{\mathbf{if}\ L<0\ \mathbf{then}\ M\ \mathbf{else}\ N}(\boldsymbol{p})(\boldsymbol{z}_1)(\boldsymbol{z}_2)(\boldsymbol{z}_3)(\underbrace{[\![N]\!]^t(\boldsymbol{f},\boldsymbol{z}_3)}_{[\![\mathbf{if}\ L<0\ \mathbf{then}\ M\ \mathbf{else}\ N]\!]^t(\boldsymbol{f},\boldsymbol{z}_1 +\!\!+ \boldsymbol{z}_2 +\!\!+ \boldsymbol{z}_3)}) = 1$$

– Suppose $\Gamma \mid [] \vdash_{\text{aff}} \lambda y.\, M : \tau_1 \bullet \Sigma_1 \to \cdots \to \tau_\ell \bullet \Sigma_\ell \to \iota^{(a)}$, where $\ell \geq 1$, and $\boldsymbol{f} \preceq' \boldsymbol{p}$. To show

$$[\![\lambda y.\, M]\!]^t(\boldsymbol{f}, []) \preceq_{\tau_1 \bullet \Sigma_1 \to \cdots \to \tau_\ell \bullet \Sigma_\ell \to \iota^{(a)}} \mathrm{Gr}^{\mathcal{A}}_{\lambda y.M}(\boldsymbol{p})$$

let $g_1 \preceq'_{\tau_1} q_1, \ldots, g_\ell \preceq'_{\tau_\ell} q_\ell$ and $\boldsymbol{z}_1 \in \mathbb{R}^{|\Sigma_1|}, \ldots, \boldsymbol{z}_\ell \in \mathbb{R}^{|\Sigma_\ell|}$.

By the inductive hypothesis, $[\![M]\!]^t \preceq \mathrm{Gr}^{\mathcal{A}}_{M}$, which implies

$$\mathrm{Gr}^{\mathcal{A}}_{M}(\boldsymbol{p})(q_1)(\boldsymbol{z}_1)\cdots(q_\ell)(\boldsymbol{z}_\ell)\left([\![M]\!]^t((\boldsymbol{f},g_1),\boldsymbol{z}_1)(g_2,\boldsymbol{z}_2)\cdots(g_\ell,\boldsymbol{z}_\ell)\right) = 1$$

Therefore, by assumption that $\mathcal{A} \models \Phi_{\lambda y.M}$,

$$\mathrm{Gr}^{\mathcal{A}}_{\lambda y.M}(\boldsymbol{p})(q_1)(\boldsymbol{z}_1)\cdots(q_\ell)(\boldsymbol{z}_\ell)\left(\underbrace{[\![M]\!]^t(\boldsymbol{f},g_1,\boldsymbol{z}_1)}_{[\![\lambda y.M]\!]^t(\boldsymbol{f},[])(g_1,\boldsymbol{z}_1)}(g_2,\boldsymbol{z}_2)\cdots(g_\ell,\boldsymbol{z}_\ell)\right) = 1$$

– Suppose $\Gamma \mid \Sigma_1 +\!\!+ \Sigma_2 +\!\!+ \Sigma_3 \vdash_{\text{aff}} M\,N : \tau_2 \bullet \Sigma'_2 \to \cdots \to \tau_\ell \bullet \Sigma'_\ell \to \iota^{(a)}$ because $\Gamma \mid \Sigma_1 \vdash_{\text{aff}} M : \tau_1 \bullet \Sigma_3 \to \tau_2 \bullet \Sigma'_2 \to \cdots \to \tau'_\ell \bullet \Sigma'_\ell \to \iota^{(a)}$ and $\Gamma \mid \Sigma_2 \vdash_{\text{aff}} N : \tau_1$, and let $\boldsymbol{f} \preceq' \boldsymbol{p}$. To show

$$[\![M\,N]\!]^t(\boldsymbol{f},\boldsymbol{z}_1 +\!\!+ \boldsymbol{z}_2 +\!\!+ \boldsymbol{z}_3) \preceq_{\tau_2 \bullet \Sigma'_2 \to \cdots \to \tau_\ell \bullet \Sigma'_\ell \to \iota^{(a)}} \mathrm{Gr}^{\mathcal{A}}_{M\,N}(\boldsymbol{p})(\boldsymbol{z}_1)(\boldsymbol{z}_2)(\boldsymbol{z}_3)$$

let $g_2 \preceq'_{\tau'_2} q_2, \ldots g_\ell \preceq'_{\tau'_\ell} q_\ell$ and $\boldsymbol{z}'_2 \in \mathbb{R}^{|\Sigma'_2|}, \ldots, \boldsymbol{z}'_\ell \in \mathbb{R}^{|\Sigma'_\ell|}$. If $\tau_1 = \iota^{(a)}$ then by the inductive hypothesis,

$$\mathrm{Gr}^{\mathcal{A}}_{N}(\boldsymbol{p})(\boldsymbol{z}_2)\left([\![N]\!]^t(\boldsymbol{f},\boldsymbol{z}_2)\right) = 1$$
$$\mathrm{Gr}^{\mathcal{A}}_{M}(\boldsymbol{p})(\boldsymbol{z}_1)([\![N]\!]^t(\boldsymbol{f},\boldsymbol{z}_2))(\boldsymbol{z}_3)(q_2)(\boldsymbol{z}'_2)\cdots(q_\ell)(\boldsymbol{z}'_\ell)$$
$$\left([\![M]\!]^t(\boldsymbol{f},\boldsymbol{z}_1)([\![N]\!]^t(\boldsymbol{f},\boldsymbol{z}_2),\boldsymbol{z}_3)(g_2,\boldsymbol{z}'_2)\cdots(g_\ell,\boldsymbol{z}'_\ell)\right) = 1$$

and due to $\mathcal{A} \models \Phi_{M\,N}$,

$$\mathrm{Gr}^{\mathcal{A}}_{M\,N}(\boldsymbol{p})(\boldsymbol{z}_1)(\boldsymbol{z}_2)(\boldsymbol{z}_3)(q_2)(\boldsymbol{z}'_2)\cdots(q_\ell)(\boldsymbol{z}'_\ell)$$
$$\left(\underbrace{[\![M]\!]^t(\boldsymbol{f},\boldsymbol{z}_1)([\![N]\!]^t(\boldsymbol{f},\boldsymbol{z}_2),\boldsymbol{z}_3)}_{[\![M\,N]\!]^t(\boldsymbol{f},\boldsymbol{z}_1 +\!\!+ \boldsymbol{z}_2 +\!\!+ \boldsymbol{z}_3)}(g_2,\boldsymbol{z}'_2)\cdots(g_\ell,\boldsymbol{z}'_\ell)\right) = 1$$

Otherwise, by the inductive hypothesis,

$$\mathrm{Gr}^{\mathcal{A}}_{N}(\boldsymbol{p})(\boldsymbol{z}_2) \preceq_{\tau_1} [\![N]\!]^t(\boldsymbol{f},\boldsymbol{z}_2)$$

and the claim follows similarly by definition of $\preceq$. □

Index

Conventions

$[\varphi]$	Iverson bracket
id	identity function
∂U	boundary of U
$\mathring{U}$	interior of U
$\overline{U}$	closure of U
$\mathbf{B}_r(\mathbf{x})$	ball of radius $r > 0$ centred at $\mathbf{x}$
$\mathbf{e}_i$	i-th standard basis unit vector
$\mathbf{J}$	Jacobian matrix
det	determinant
$\frac{\partial f}{\partial x}$	partial derivative of f
∇f	gradient of f
μ	measures
Leb	Lebesgue measure
$\mathcal{D}, \mathcal{D}_{\boldsymbol{\theta}}$	distributions
$\boldsymbol{\mathcal{D}}, \boldsymbol{\mathcal{D}}_{\boldsymbol{\theta}}$	multivariate distributions
$\mathrm{supp}(\boldsymbol{\mathcal{D}})$	support
$\mathcal{N}(\mu, \sigma^2)$	normal distributions
$\mathcal{N}$	standard normal distribution
$\mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\mathcal{D}}}[f(\boldsymbol{x})]$	expectations
$\boldsymbol{\theta}$	parameters
$\boldsymbol{\Theta}$	parameter space
$\boldsymbol{s}$	samples
$\boldsymbol{z}$	transformed samples
$\boldsymbol{\varphi}_{\boldsymbol{\theta}}$	diffeomorphic transformations
Op	primitive operations
Dist	primitive probability distributions
$\mathrm{Diff}_{\boldsymbol{\Theta}}$	diffeomorphic transformations
L, M, N	terms
τ	types
ι	base types/sorts
Γ	contexts
R	type/sort of reals
$R_{>0}$	type/sort of positive reals
Σ	trace types
V	values
R	redexes
E	evaluation contexts
Λ	set of terms
Λ^0	set of closed terms