MPRI – Probabilistic Programming Languages
Discrete semantics

Hugo Paquet

17 Oct 2025

1 Semantics and equational theory

1. Give a type and compute the semantics of the following two terms:

```
let x = bernoulli 0.3 in
let y = bernoulli 0.4 in
assume (x == y);
return x
```

```
(λt. let x = t () in (let y = t () in return (x + y)))(λz. bernoulli 0.5)
```

2. Show that for all programs s, t, u of the language with $\Gamma \vdash^{\mathrm{P}} s : A$, $\Gamma \vdash^{\mathrm{P}} t : B$ and $\Gamma, x : A, y : B \vdash^{\mathrm{P}} u : C$, the following two programs have the same semantics:

```
let x = s in
let y = t in
u
```

=

```
let y = t in
let x = s in
u
```

3. Show that moreover the following equation holds for terms $\Gamma \vdash^{\mathrm{P}} s : A$ and $\Gamma \vdash^{\mathrm{D}} t : B$ whenever the program s contains no observations (=no instances of `assume`).

```
let x = s in
t
```

=

```
t
```

Give an example (using `assume`) for which the equation does not hold.

2 Monads and effects

The previous section says that dist and $\text{dist}_{\leq}$ are *commutative monads* (Question 2), and that dist (but not $\text{dist}_{\leq}$) is an *affine* monad.

4. Give the return and bind structure for the exception monad $X \mapsto X + 1$.

5. Let S be a set. We think of S as a set of possible memory states. Give the return and bind structure for a state monad $X \mapsto S \to (X \times S)$ modelling programs which can read and modify the memory. Then give a combined monad for probabilistic programs which can do both kinds of effects.

6. Are the monads of this section commutative or affine?

3 Discrete measure theory

7. Show that every function between discrete measurable spaces is a measurable function.

8. Show that a probability measure μ on a discrete measurable space $(X, \mathcal{P}X)$ is completely determined by its value on singleton sets.