

Probabilistic Sub-Types

by Sven Nilsen, 2019

In this paper I introduce a notation for probabilistic sub-types.

Probabilistic sub-types is a generalization of notation for sub-types which truth value is `true` or `false`. Instead of using a boolean as a truth value, one uses probabilities. Formally, a probabilistic sub-type is a statement that has a truth value in the unit interval $[0, 1]$ of real numbers:

$$[f]_p a : \text{prob}$$

$$\text{prob} := \lambda(x : \text{real}) = (x \geq 0) \wedge (x \leq 1)$$

The interpretation of sub-types which truth value is a boolean is referred to as “boolean sub-types”. To keep probabilistic sub-types compatible with the notation for boolean sub-types, a sub-script `p` is written after the square bracket. This means that the sub-type is interpreted as a probability.

$[f]_p a$	$=$	$(\exists_p f)(a)$	The probability that `f` returns `a`
$\neg[f]_p a$	$=$	$1 - (\exists_p f)(a)$	The probability that `f` does not return `a`

When there is no sub-script `p` after the square bracket, the following interpretation is used:

$$\begin{aligned} [f] a &\leq \Rightarrow [f]_p a > 0 \\ [f] a &= a : [\exists_p f] (> 0) \end{aligned}$$

This interpretation does not terminate, but converges to some boolean value. If you are familiar with boolean type-checking in path semantics, you might know that double-existential paths $\exists\exists f$ is one of four functions of type $\text{bool} \rightarrow \text{bool}$. It is not possible to check the consistency of a statement without realizing the fact that this goes on forever but in a predictable and repeating pattern. Similarly, in probabilistic path semantics, the interpretation of a boolean sub-type goes on forever, but it is much more complex when defined in terms of the probabilistic existential path. By realizing the fact that consistency of probabilistic sub-types converges to checks predicted by boolean sub-types, one can stop execution and use the rules for checking boolean sub-types instead.

When mixing boolean sub-types with probabilistic sub-types, the boolean sub-types are used as domain constraints on the probabilistic existential path:

$$[g] b \wedge [f]_p a = (\exists_p f \{ [g] b \})(a)$$

The above means the probability that `f` returns `a`, given that `g` returns `b`.

This can also be written in the familiar notation conditional probability:

$$P([f] a \mid [g] b)$$

Conditional probabilities share the semantics of mixing boolean sub-types and probabilistic sub-types.