Uniform Properties of Sets

by Sven Nilsen, 2018

A uniform property of a set is a function of type `set → bool` that returns `true` for every subset of the set for which it returns `true`. A uniform property satisfies the sub-type `uniform_set_property`:

p : uniform_set_property `p` is a uniform property of a set

uniform_set_property := \(p : set → bool) = ∀ s : p, s' : 2^s { p(s') }

The operation `$2^s$` means taking the powerset of the set `s`. There are two for-all loops.

Written in de-sugared form:

p : [uniform_set_property] true

uniform_set_property := \(p : set → bool) = ∀ s : p { ∀ s' : powerset(s) { p(s') } }

Another form:

uniform_set_property := \(p : set → bool) = ∀ s : set { ∀ s' : set { p(s) ∧ s' ⊆ s → p(s') } }

When a set `s` satisfies the property `p` one can write:

s : p

Which is the same as:

s : [p] true

A general example of uniform property is any statement said for all members in a set that only depends on that specific member. For example, when all members of a set are blue, determined by an `all_blue` function, then no matter how we remove members, the set of the remaining members will only contain blue members. It is a uniform set property:

all_blue : uniform_set_property

all_blue := \(s : set) = ∀ m : s { blue(m) }

Another example from geometry: The set of all points on a straight line has zero distance to the straight line. If we construct new lines from a sub-set of these points, then all the new lines will lie on the same straight line. This is kind of trivial.

There are non-trivial cases too: If every person in a city has two siblings living in the same city, then this is NOT a uniform set property since the city must contain at least 3 people. On the other hand, if every person in a city is unique in some way, then it is a uniform set property.