The following is an attempt to write a denotational semantics for Nu's scripting system. This is a WIP.

µ:Axiom α = String -> α

µ:Axiom' α \_ = String -> α

µ:Value α = µ:Axiom α "A primitive value."

µ:Address = µ:Axiom Unit "A locator for an active resource, such as an event or simulant."

µ:Relation = µ:Address -> µ:Address

µ:Stream α β γ = Lambda α β -> Lambda β γ

µ:Constant α = Name -> µ:Value α

µ:Variable α β γ = Name -> µ:Stream α β γ

µ:Variable' α β γ = Name -> µ:Relation (ctx ()) -> µ:Stream α β γ (Lambda α β)

µ:Equality α β γ = µ:Stream α β γ (Lambda α β) (\x => (µ:Command γ x (µ:Set Name (µ:Relation (ctx ())))))

µ:Handler α β γ = µ:Stream α β γ (µ:Command β)

µ:Command α = Value α ->

| µ:Get = Name -> Address -> α

| µ:Set = Name -> Address -> Unit

| µ:\_ = Name -> α

ctx () : Address = addr

runStream α β γ x stream : γ = µ:Axiom' γ x "Executes the linked lambdas in the stream."

lambdizeStream α β γ stream : Lambda α γ = \x -> runStream x stream

product stream stream2 : µ:Stream α β γ = \x -> \y -> (runStream x stream, runStream y stream2)