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

µ:Axiom *a* = String -> *a*

µ:Axiom' *a* \_ = String -> *a*

µ:Effect = µ:Axiom Unit "An effect on the state of the world."

µ:Value *a* = µ:Axiom *a* "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 *a* *b* *c* = Lambda *a* *b* -> Lambda *b* *c*

µ:Constant *a* = Name -> µ:Value *a*

µ:Variable *a* *b* *c* = Name -> µ:Stream *a* *b* *c*

µ:Variable' *a* *b* *c* = Name -> µ:Relation (ctx ()) -> µ:Stream *a* *b* *c* (Lambda *a* *b*)

µ:Equality *a* *b* *c* = µ:Stream *a* *b* *c* (Lambda *a* *b*) (\x => (µ:Command *c* x (µ:Set Name (µ:Relation (ctx ())))))

µ:Handler *a* *b* *c* = µ:Stream *a* *b* *c* (µ:Command *b*)

µ:Command *a* = Value *a* ->

| µ:Get = Name -> Address -> *a*

| µ:Set = Name -> Address -> Effect

| µ:\_ = Name -> *a*

ctx () : Address = addr

runStream *a* *b* *c* x stream : *c* = µ:Axiom' *c* x "Executes the linked lambdas in the stream."

lambdizeStream *a* *b* *c* stream : Lambda *a* *c* = \x -> runStream x stream

product stream stream2 : µ:Stream *a* *b* *c* = \x -> \y -> (runStream x stream, runStream y stream2)