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

For more info, look here - <https://www.youtube.com/watch?v=bmKYiUOEo2A>

mu :: *a* Axiom -> String -> *a*

mu :: *a* Value -> *a* Axiom "a primitive value."

mu :: *a* Sequence -> *a* Axiom "an ordered sequence of things."

mu :: Address -> Unit Axiom "locator for an active resource, such as an event or simulant."

mu :: *a* Command =

| *a* -> Name -> Unit

| *a* Get -> Name -> Address -> *a*

| \v -> *a* Set v -> Name -> Address -> Unit

mu :: Any = AllowAny | DisallowAny

mu :: *a* Relation -> Address -> Address

mu :: *a* *b* Stream -> *b* Value Sequence

mu :: *a* Constant -> Name -> *a* Value

mu :: *a* *b* Variable -> Name -> *a* *b* Stream

mu :: *a* *b* Variable' -> Name -> DisallowAny Relation (ctx ()) -> *a* *b* Stream

mu :: Equality -> Name -> AllowAny Relation (ctx ()) -> *a* *b* Stream -> *b* Set Command

mu :: Handler -> *a* *b* Stream -> *b* Command

*where*

mu :: ctx () = addr

mu :: liveness () = Bool Value