

The following is an attempt to write a denotational semantics for Nu's scripting system, based on <https://www.youtube.com/watch?v=bmKYiUOEo2A>. This presentation uses a more improvised style of syntax than the Conal's, however.

Axiomatic semantics - When a  $\mu$  is defined in terms of itself, we consider it axiomatic and irreducible in this context.

```
 $\mu$ :Value<a> =  
|  $\mu$ :Value< $\mu$ :Relation>  
|  $\mu$ :Value< $\mu$ :Address>  
|  $\mu$ :Value< $\mu$ :Name>  
|  $\mu$ :Value< $\mu$ :String>  
|  $\mu$ :Value< $\mu$ :Bool>  
|  $\mu$ :Value< $\mu$ :Unit>
```

```
 $\mu$ :Values<a> =  $\mu$ :Values<a>
```

```
 $\mu$ :Effect =  $\mu$ :Effect
```

### Derived Semantics

```
 $\mu$ :Get<a> =  $\mu$ :Name ->  $\mu$ :Relation ->  $\mu$ :Value<a>  
 $\mu$ :Set<a> =  $\mu$ :Name ->  $\mu$ :Relation ->  $\mu$ :Value<a> ->  $\mu$ :Effect  
 $\mu$ :Cmd<a> =  $\mu$ :Value<a> ->  $\mu$ :Effect  
 $\mu$ :Fold<a b> = ( $\mu$ :Value<a> -> b) ->  $\mu$ :Values<a> -> b
```

```
 $\mu$ :Define<a> =  
|  $\mu$ :Name ->  $\mu$ :Value<a>  
|  $\mu$ :Name ->  $\mu$ :Name ->  $\mu$ :Relation ->  $\mu$ :Value<a>
```

```
 $\mu$ :Stream<a> =  
|  $\mu$ :Values<a>  
|  $\mu$ :Address ->  $\mu$ :Values<a>  
|  $\mu$ :Name ->  $\mu$ :Relation ->  $\mu$ :Values<a>  
|  $\mu$ :Stream<a> -> (a -> b) ->  $\mu$ :Values<b>  
|  $\mu$ :Stream<a> ->  $\mu$ :Stream<b> ->  $\mu$ :Stream (a * b)  
|  $\mu$ :Stream<a> ->  $\mu$ :Stream<b> ->  $\mu$ :Stream (a | b)
```

```
 $\mu$ :Variable<a> =  $\mu$ :Name ->  $\mu$ :Stream a
```

```
 $\mu$ :Equate<a> = \name -> \rel ->  $\mu$ :Stream<a> ->  $\mu$ :Fold ( $\mu$ :Set<a> name rel)
```

```
 $\mu$ :Handle<a> =  $\mu$ :Stream<a> ->  $\mu$ :Fold  $\mu$ :Cmd<a>
```