Rules

Safe Haskell	None
Language	Haskell2010

This module defines the data structures, helper functions and declarations for the fuzzy logic relations which will be used in out FLC.

Documentation

```
class Factor a where
```

`Factor` is a class representing inputs which are factors in deciding the final action.

Methods

```
factor :: a -> Action
```

```
data Action
```

`Action` is the type representing the discreet set of action we might take.

Constructors

HardSlowdown
Slowdown
NoOP
Speedup
HardSpeedup

Instances

`Bounded Action`
`Enum Action`
`Eq Action`
`Ord Action`
`Show Action`

```
data Speed
```

`Speed` is the type representing the discreet set of possible speed "ratings".

Constructors

VerySlow
Slow
Normal
Fast
VeryFast

Instances

`Bounded Speed`

Enum Speed

Eq Speed

Ord Speed

Show Speed

```
data Distance
```

Distance represents the type of the discreet set of relative distances to the destination.

Constructors

VeryClose
Close
Halfway
Far
VeryFar

Instances

Bounded Distance

Enum Distance

Eq Distance

Ord Distance

Show Distance

```
type Rule = [FuzzySet]
```

Rule is a list of the degrees of activation of each premise. (i.e. foldr unionT rule == fuzzy logic rule)

```
mkRuler :: Reader Config ([(Int, Int)] -> Rule)
```

mkRuler takes a list of term limits and returns the associated Rule. It relies on the config keys `totalSpace`, `ruleDelta`.