0000

# FuzzyClips

Francisco Carrillo Pérez

Universidad de Granada

14 de mayo de 2016

### **INDICE**

Introducción

Instalación

¿Cómo  $p^{***}$  se programa en esto?

Tranquilidad

**Fuzzy Templates** 

**Fuzzy Facts** 

**Fuzzy Rules** 

**Problemas** 

Alternativas y conclusión

Bibliografía

# Overview

# Introducción

Inetalación

¿Cómo p\*\*\* se programa en esto?

Problemas

Alternativas v conclusión

Bibliografí

## Introducción

FuzzyClips es una extensión de CLIPS creada por el *Integrated Reasoning Group of the Institute for Information Technology of the National Research Council of Canada*, que nos da la capacidad de representar y manipular reglas y hechos difusos.

Introducción

## Instalación

¿Cómo p\*\*\* se programa en esto?

Problemas

Alternativas v conclusión

Bibliografí

### Instalación y Documentación

Mirror donde poder descargar los ejecutables de Windows y Linux: http://awesom.eu/~quentin/archives/2010/04/22/fuzzyclips\_downloads/index.html.

También se puede encontrar en el mismo enlace la documentación.

Introducción

Instalación

```
¿Cómo p*** se programa en esto?
Tranquilidad
Fuzzy Templates
Fuzzy Facts
Fuzzy Rules
```

Problemas

Alternativas y conclusión

# Tranquilidad

0000

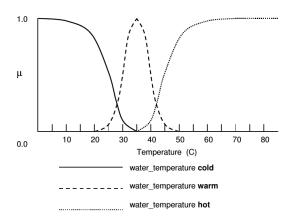
La sintáxis es la misma que en CLIPS, solo se añaden ciertas funcionalidades que es lo que nos permite la representación de la incertidumbre.

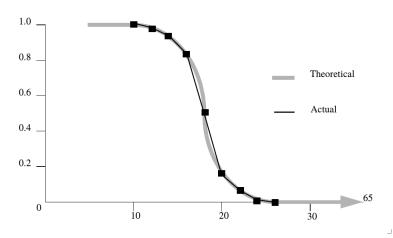
#### **DEFTEMPLATE**

Todas las variables fuzzy deben declararse en un template antes de poder usarse. Se declara el template de la siguiente forma:

## Ejemplo de como quedaría un deftemplate:

```
(deftemplate Tx "output water temperature"
   5 65 Celsius
        (cold (z 10 26)); standard set representation
        (OK (PI 2 36))
        (hot (s 37 60))
```





#### **DEFFACTS**

El formato que tendrían los deffacts sería el siguiente:

Un ejemplo simple con el template anterior sería:

0000

```
(deffacts fuzzy-fact
    (Tx cold)
```

#### **DEFRULE**

La sintaxis de los defrule y su construcción es la misma, un pequeño ejemplo sería:

```
(defrule one
    (Tx cold)
    =>
     (assert (Feeling FrioDelCarajo))
)
```

En este caso estaríamos también insertando un hecho fuzzy.

Introducción

Instalación

¿Cómo p\*\*\* se programa en esto?

### Problemas

Alternativas v conclusión

Bibliografí

### **PROBLEMAS**

- 1. No tiene mantenimiento(o no tiene pinta de que tenga).
- 2. Escasa documentación.
- 3. ¡El mirror de descarga lo tiene un estudiante(o antiguo estudiante, a saber) en su página web!

Introducción

Instalación

¿Cómo p\*\*\* se programa en esto?

Problemas

Alternativas y conclusión

Bibliografía

### Conclusión

Es una herramienta potente, pero se ve limitada por la excasa documentación. El código fuente está subido en este repo de Github: https://github.com/rorchard/FuzzyCLIPS. No hay mucha comunidad, haciéndo búsqueda en Google no encuentras mucho.

### **ALTERNATIVAS**

La principal alternativa que he encotrado es scikit-fuzzy https://github.com/scikit-fuzzy/scikit-fuzzy, que es una librería para Python  $\heartsuit$ .

Introducción

Inetalación

¿Cómo p\*\*\* se programa en esto?

Problemas

Alternativas v conclusión

Bibliografía

# Bibliografía

- ► http://awesom.eu/~quentin/archives/2010/04/22/ fuzzyclips\_downloads/index.html
- ▶ http://alumni.cs.ucr.edu/~vladimir/cs171/ quickfuzzy.pdf

0000