

# Implementación de técnicas de despliegado difuso sobre el entorno FLOPER

José Antonio Riaza Valverde

## Trabajo Fin de Grado

Grado en Ingeniería Informática, Computación  
Universidad de Castilla-La Mancha

**Directores:** Ginés Damián Moreno Valverde  
Jaime Penabad Vázquez

27 de junio de 2017

- 1 Motivación y Objetivos
- 2 Lenguaje FASILL
  - Sintaxis
  - Semántica operacional
  - El entorno FLOPER online
- 3 Despliegado de programas difusos
  - Definición
  - Ejemplo
  - Despliegado en el entorno FLOPER online
- 4 Lenguaje sFASILL
  - Sintaxis
  - Semántica operacional
  - Despliegado de programas simbólicos
- 5 Calibrado de programas simbólicos
  - Definición
  - Ejemplo
  - Calibrado en el entorno FLOPER online
- 6 Conclusiones y trabajo futuro

- 1 Motivación y Objetivos
- 2 Lenguaje FASILL
  - Sintaxis
  - Semántica operacional
  - El entorno FLOPER online
- 3 Despliegado de programas difusos
  - Definición
  - Ejemplo
  - Despliegado en el entorno FLOPER online
- 4 Lenguaje sFASILL
  - Sintaxis
  - Semántica operacional
  - Despliegado de programas simbólicos
- 5 Calibrado de programas simbólicos
  - Definición
  - Ejemplo
  - Calibrado en el entorno FLOPER online
- 6 Conclusiones y trabajo futuro

# Motivación y Objetivos

- **FLOPER**

Entorno de programación lógica difusa desarrollado en la UCLM por el grupo DEC-TAU.

- **MALP**

- Retículos multi-adjuntos.

- **FASILL**

- Retículos completos.

- Relaciones de similaridad.

- **Objetivos**

- 1 Despliegado de programas difusos.
- 2 Programas lógicos difusos simbólicos.
- 3 Calibrado de programas simbólicos.
- 4 Integración en la herramienta online.

- 1 Motivación y Objetivos
- 2 Lenguaje FASILL
  - Sintaxis
  - Semántica operacional
  - El entorno FLOPER online
- 3 Despliegado de programas difusos
  - Definición
  - Ejemplo
  - Despliegado en el entorno FLOPER online
- 4 Lenguaje sFASILL
  - Sintaxis
  - Semántica operacional
  - Despliegado de programas simbólicos
- 5 Calibrado de programas simbólicos
  - Definición
  - Ejemplo
  - Calibrado en el entorno FLOPER online
- 6 Conclusiones y trabajo futuro

# Lenguaje FASILL

FASILL (*Fuzzy Aggregators and Similarity Into a Logic Language*) es un lenguaje de programación **lógico difuso** que trabaja con **retículos completos** y permite **relaciones de similitud**.

## Programa FASILL

Un **programa FASILL**  $\mathcal{P}$  es una 3-tupla  $\langle \Pi, \mathcal{R}, L \rangle$ , donde  $\Pi$  es un conjunto de reglas FASILL,  $\mathcal{R}$  es una relación de similitud y  $L$  es un retículo completo.

# Reglas

## Regla FASILL

Una **regla FASILL** es de la forma  $A \leftarrow B \in \Pi$ , donde  $A$  (la cabeza) es una fórmula atómica y  $B$  (el cuerpo) es una fórmula bien formada, construida a partir de fórmulas atómicas  $B_1, B_2, \dots, B_n$ , grados de verdad y conectivas de  $L$ .

$R_1 : \text{vanguardist}(\text{hydropolis}) \leftarrow 0.9.$   
 $R_2 : \text{elegant}(\text{ritz}) \leftarrow 0.8.$   
 $R_3 : \text{close}(\text{hydropolis}, \text{taxi}) \leftarrow 0.7.$   
 $R_4 : \text{good\_hotel}(X) \leftarrow @_{\text{aver}}(\text{elegant}(X), @_{\text{very}}(\text{close}(X, \text{metro}))).$

$\mathcal{R}(\text{elegant}, \text{vanguardist}) = 0.6$      $\mathcal{R}(\text{metro}, \text{taxi}) = 0.4$   
 $\mathcal{R}(\text{metro}, \text{bus}) = 0.5$                      $\mathcal{R}(\text{taxi}, \text{bus}) = 0.4$

# Retículo completo

## Retículo completo

Un **retículo completo** es un conjunto ordenado  $(L, \leq)$  tal que todo subconjunto  $S$  de  $L$  tiene  $\inf(S)$  y  $\sup(S)$ . En particular, existe el ínfimo de  $L$  y el supremo de  $L$ , denotados por  $\perp$  y  $\top$ , respectivamente.

El retículo  $L$  está equipado con un conjunto de conectivas  $\varsigma$ :

- **t-normas** y **t-conormas**, denotadas por  $\&$  y  $|$ .
- **agregadores**, denotados por  $@$ .

En esta presentación utilizaremos el retículo  $L = ([0, 1], \leq)$ , donde  $\leq$  es el orden usual.

## Relación de similaridad

### Relación de similaridad

Dado un dominio  $\mathcal{U}$  y un retículo  $L$  con una t-norma  $\wedge$  fija, una **relación de similaridad**  $\mathcal{R}$  es una relación binaria difusa sobre  $\mathcal{U}$  en  $L$ , que satisface las siguientes propiedades:

- Reflexiva:  $\mathcal{R}(x, x) = \top, \forall x \in \mathcal{U}$ .
- Simétrica:  $\mathcal{R}(x, y) = \mathcal{R}(y, x), \forall x, y \in \mathcal{U}$ .
- Transitiva:  $\mathcal{R}(x, z) \geq \mathcal{R}(x, y) \wedge \mathcal{R}(y, z), \forall x, y, z \in \mathcal{U}$ .

## Relación de similaridad

La relación de similaridad  $\mathcal{R}$  sobre  $\mathcal{U} = \{vanguardist, elegant, metro, taxi, bus\}$  y  $L = ([0, 1], \leq)$  se expresa mediante la siguiente matriz:

$\mathcal{R}$	vanguardist	elegant	metro	taxi	bus
vanguardist	1	0.6	0	0	0
elegant	0.6	1	0	0	0
metro	0	0	1	0.4	0.5
taxi	0	0	0.4	1	0.4
bus	0	0	0.5	0.4	1

$$\mathcal{R}(\text{metro}, \text{taxi}) \geq \mathcal{R}(\text{metro}, \text{bus}) \wedge \mathcal{R}(\text{bus}, \text{taxi}) = 0.5 \wedge 0.4$$

# Unificación débil

## Unificación débil

Dos expresiones  $\mathcal{E}_1$  y  $\mathcal{E}_2$  unifican débilmente si  $\mathcal{R}(\mathcal{E}_1, \mathcal{E}_2) = r > \perp$ ; en ese caso, se dice que  $wmgu(\mathcal{E}_1, \mathcal{E}_2) = \langle \theta, r \rangle$ .

$$\begin{aligned} \langle \{ \text{elegant}(\text{taxi}) \approx \text{vanguardist}(\text{metro}) \}, id, 1 \rangle &\implies \\ \langle \{ \text{taxi} \approx \text{metro} \}, id, 0.6 \rangle &\implies \\ \langle \{ \}, id, 0.6 \wedge 0.4 \rangle = \langle \{ \}, id, 0.4 \rangle \end{aligned}$$

$$\begin{aligned} \langle \{ \text{elegant}(\text{taxi}) \approx \text{vanguardist}(X) \}, id, 1 \rangle &\implies \\ \langle \{ \text{taxi} \approx X \}, id, 0.6 \rangle \implies \langle \{ X \approx \text{taxi} \}, id, 0.6 \rangle &\implies \\ \langle \{ \}, \{ X / \text{taxi} \}, 0.6 \rangle \end{aligned}$$

## Pasos de computación

SS  
↪

**Paso de éxito:**

$$\frac{\langle Q[A]; \sigma \rangle \quad A' \leftarrow B \in \Pi \quad \text{wmgu}(A, A') = \langle \theta; r \rangle}{\langle Q[A/B \wedge r]\theta; \sigma\theta \rangle} \text{SS}$$

FS  
↪

**Paso de fallo:**

$$\frac{\langle Q[A]; \sigma \rangle \quad \nexists A' \leftarrow B \in \Pi : \text{wmgu}(A, A') = \langle \theta; r \rangle, r > \perp}{\langle Q[A/\perp]; \sigma \rangle} \text{FS}$$

IS  
↪

**Paso interpretativo:**

$$\frac{\langle Q[\zeta(r_1, \dots, r_n)]; \sigma \rangle \quad \dot{\zeta}(r_1, \dots, r_n) = r_{n+1}}{\langle Q[\zeta(r_1, \dots, r_n)/r_{n+1}]; \sigma \rangle} \text{IS}$$

# Pasos de computación

$$\Pi = \begin{cases} \mathbf{R}_1 : \text{vanguardist}(\text{hydropolis}) & \leftarrow 0.9. \\ \mathbf{R}_2 : \text{elegant}(\text{ritz}) & \leftarrow 0.8. \\ \mathbf{R}_3 : \text{close}(\text{hydropolis}, \text{taxi}) & \leftarrow 0.7. \\ \mathbf{R}_4 : \text{good\_hotel}(X) & \leftarrow @_{\text{aver}}(\text{elegant}(X), \\ & @_{\text{very}}(\text{close}(X, \text{metro}))). \end{cases}$$

- $\langle \text{elegant}(X); id \rangle \xrightarrow{SS} \langle \&_{\text{godel}}(0.9, 0.6); \{X / \text{hydropolis}\} \rangle$
- $\langle \text{close}(\text{ritz}, \text{taxi}); id \rangle \xrightarrow{FS} \langle 0; id \rangle$
- $\langle @_{\text{very}}(0.8); id \rangle \xrightarrow{IS} \langle 0.64; id \rangle$

# El entorno FLOPER online

FLOPER

[Documentation](#) [Downloads](#) [Try it online](#)

## Try it!

### Program

```
vanguardist(hydropolis) <- 0.9.  
elegant(ritz) <- 0.8.  
close(hydropolis, taxi) <- 0.7.  
good_hotel(X) <- @aver(elegant(X), @very(close(X, metro))).
```

[symbolic.fpl](#) [lists.fpl](#) [good\\_hotel.fpl](#)

### Lattice

```
% Elements  
member(X) :- number(X), 0 =< X, X =< 1.  
members([0.3, 0.5, 0.7]).
```

[bool.lat.pl](#) [num.lat.pl](#)

```
% Distance  
distance(X,Y,Z) :- Z is abs(Y-X).
```

### Similarity equations

```
elegant/1 ~ vanguardist/1 = 0.6.  
metro ~ bus = 0.5.  
bus ~ taxi = 0.4.  
~tnorm = godel.
```

[good\\_hotel.sim](#)

# El entorno FLOPER online

Goal	Tree depth	Mode	Answers & Tree
good_hotel(X)	12	<span>Small</span> <span>Medium</span> <span>Large</span>	<span>Generate</span>

## Answers

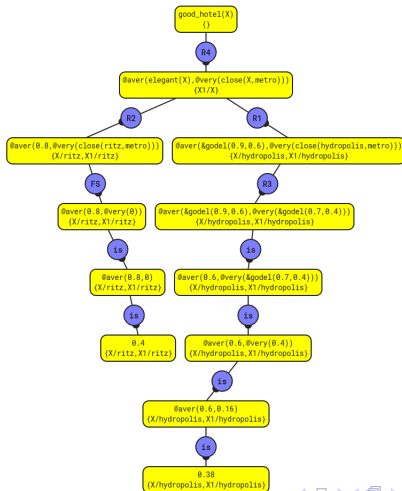
Computed answers

```
< #s3(#s2,0), {X/ritz} >  
< #s3(&godel(#s1,0.6),0.16), {X/hydropolis} >
```

Derivation tree

```
R0 < good_hotel(X), {} >  
  R4 < #s3(elegant(X),@very(close(X,metro))), {X1/X} >  
    R2 < #s3(#s2,@very(close(ritz,metro))), {X/ritz,X1/ritz} >  
      FS < #s3(#s2,@very(0)), {X/ritz,X1/ritz} >  
        is < #s3(#s2,0), {X/ritz,X1/ritz} >
```

# El entorno FLOPER online



- 1 Motivación y Objetivos
- 2 Lenguaje FASILL
  - Sintaxis
  - Semántica operacional
  - El entorno FLOPER online
- 3 Despliegado de programas difusos
  - Definición
  - Ejemplo
  - Despliegado en el entorno FLOPER online
- 4 Lenguaje sFASILL
  - Sintaxis
  - Semántica operacional
  - Despliegado de programas simbólicos
- 5 Calibrado de programas simbólicos
  - Definición
  - Ejemplo
  - Calibrado en el entorno FLOPER online
- 6 Conclusiones y trabajo futuro

## Despliegado FASILL

El objetivo de las transformaciones de despliegado es generar programas más eficientes preservando su semántica, mediante la aplicación de pasos de computación sobre sus reglas.

### Despliegado basado en similitud

Sea  $\mathcal{P} = \langle \Pi, \mathcal{R}, L \rangle$  un programa FASILL y sea  $R : A \leftarrow B \in \Pi$  una regla del programa no unitaria. Entonces, el despliegado basado en similitud de la regla  $R$  en el programa  $\mathcal{P}$  es un nuevo programa  $\mathcal{P}' = (\mathcal{P} - \{R\}) \cup \{A\sigma \leftarrow B' \mid \langle B; id \rangle \rightsquigarrow \langle B'; \sigma \rangle\}$ .

## Ejemplo ( $\Pi_1$ )

$$\Pi_0 = \left\{ \begin{array}{ll} R_1 : & \text{vanguardist(hydropolis)} \quad \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} \quad \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} \quad \leftarrow 0.7. \\ R_4 : & \text{good\_hotel(X)} \quad \leftarrow @_{\text{aver}}(\text{elegant(X)}, \\ & \quad @_{\text{very}}(\text{close(X,metro)})). \end{array} \right.$$

$$\Pi_1 = \left\{ \begin{array}{ll} R_1 : & \text{vanguardist(hydropolis)} \quad \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} \quad \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} \quad \leftarrow 0.7. \\ R_{4-1} : & \text{good\_hotel(hydropolis)} \quad \leftarrow @_{\text{aver}}(&g_{\text{odel}}(0.9,0.6), \\ & @_{\text{very}}(\text{close(hydropolis,metro)})). \\ R_{4-2} : & \text{good\_hotel(ritz)} \quad \leftarrow @_{\text{aver}}(&g_{\text{odel}}(0.8,1), \\ & @_{\text{very}}(\text{close(ritz,metro)})). \end{array} \right.$$

## Ejemplo ( $\Pi_2$ )

$$\Pi_1 = \left\{ \begin{array}{ll} R_1 : & \text{vanguardist(hydropolis)} \quad \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} \quad \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} \quad \leftarrow 0.7. \\ R_{4-1} : & \text{good\_hotel(hydropolis)} \quad \leftarrow @_{\text{aver}}(&g_{\text{godel}}(0.9,0.6), \\ & @_{\text{very}}(\text{close(hydropolis,metro)})). \\ R_{4-2} : & \text{good\_hotel(ritz)} \quad \leftarrow @_{\text{aver}}(&g_{\text{godel}}(0.8,1), \\ & @_{\text{very}}(\text{close(ritz,metro)})). \end{array} \right.$$

$$\Pi_2 = \left\{ \begin{array}{ll} R_1 : & \text{vanguardist(hydropolis)} \quad \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} \quad \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} \quad \leftarrow 0.7. \\ R_{4-1-3} : & \text{good\_hotel(hydropolis)} \quad \leftarrow @_{\text{aver}}(&g_{\text{godel}}(0.9,0.6), \\ & @_{\text{very}}(&g_{\text{godel}}(0.7,0.4))). \\ R_{4-2} : & \text{good\_hotel(ritz)} \quad \leftarrow @_{\text{aver}}(&g_{\text{godel}}(0.8,1), \\ & @_{\text{very}}(\text{close(ritz,metro)})). \end{array} \right.$$

## Ejemplo ( $\Pi_3$ )

$$\Pi_2 = \left\{ \begin{array}{ll} R_1 : & \text{vanguardist(hydropolis)} \quad \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} \quad \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} \quad \leftarrow 0.7. \\ R_{4-1-3} : & \text{good\_hotel(hydropolis)} \quad \leftarrow @\text{aver}(\&\text{godel}(0.9,0.6), \\ & \quad @\text{very}(\&\text{godel}(0.7,0.4))). \\ R_{4-2} : & \text{good\_hotel(ritz)} \quad \leftarrow @\text{aver}(\&\text{godel}(0.8,1), \\ & \quad @\text{very}(\text{close(ritz,metro)})). \end{array} \right.$$

$$\Pi_3 = \left\{ \begin{array}{ll} R_1 : & \text{vanguardist(hydropolis)} \quad \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} \quad \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} \quad \leftarrow 0.7. \\ R_{4-1-3/} : & \text{good\_hotel(hydropolis)} \quad \leftarrow @\text{aver}(0.6, \\ & \quad @\text{very}(\&\text{godel}(0.7,0.4))). \\ R_{4-2} : & \text{good\_hotel(ritz)} \quad \leftarrow @\text{aver}(\&\text{godel}(0.8,1), \\ & \quad @\text{very}(\text{close(ritz,metro)})). \end{array} \right.$$

## Ejemplo ( $\Pi_4$ )

$$\Pi_3 = \left\{ \begin{array}{lll} R_1 : & \text{vanguardist(hydropolis)} & \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} & \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} & \leftarrow 0.7. \\ R_{4-1-3/} : & \text{good\_hotel(hydropolis)} & \leftarrow @\text{aver}(0.6, \\ & & @\text{very}(\&\text{godel}(0.7,0.4))). \\ R_{4-2} : & \text{good\_hotel(ritz)} & \leftarrow @\text{aver}(\&\text{godel}(0.8,1), \\ & & @\text{very}(\text{close(ritz,metro)})). \end{array} \right.$$
  
$$\Pi_4 = \left\{ \begin{array}{lll} R_1 : & \text{vanguardist(hydropolis)} & \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} & \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} & \leftarrow 0.7. \\ R_{4-1-3//} : & \text{good\_hotel(hydropolis)} & \leftarrow @\text{aver}(0.6,@\text{very}(0.4)). \\ R_{4-2} : & \text{good\_hotel(ritz)} & \leftarrow @\text{aver}(\&\text{godel}(0.8,1), \\ & & @\text{very}(\text{close(ritz,metro)})). \end{array} \right.$$

## Ejemplo ( $\Pi_5$ )

$$\Pi_4 = \left\{ \begin{array}{lll} R_1 : & \text{vanguardist(hydropolis)} & \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} & \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} & \leftarrow 0.7. \\ R_{4-1-3//} : & \text{good\_hotel(hydropolis)} & \leftarrow @\text{aver}(0.6, @\text{very}(0.4)). \\ R_{4-2} : & \text{good\_hotel(ritz)} & \leftarrow @\text{aver}(\&\text{godel}(0.8,1), \\ & & @\text{very}(\text{close(ritz,metro)})). \end{array} \right.$$

$$\Pi_5 = \left\{ \begin{array}{lll} R_1 : & \text{vanguardist(hydropolis)} & \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} & \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} & \leftarrow 0.7. \\ R_{4-1-3//} : & \text{good\_hotel(hydropolis)} & \leftarrow @\text{aver}(0.6, 0.16). \\ R_{4-2} : & \text{good\_hotel(ritz)} & \leftarrow @\text{aver}(\&\text{godel}(0.8,1), \\ & & @\text{very}(\text{close(ritz,metro)})). \end{array} \right.$$

## Ejemplo ( $\Pi_6$ )

$$\Pi_5 = \left\{ \begin{array}{lll} R_1 : & \text{vanguardist(hydropolis)} & \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} & \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} & \leftarrow 0.7. \\ R_{4-1-3} : & \text{good\_hotel(hydropolis)} & \leftarrow @\text{aver}(0.6,0.16). \\ R_{4-2} : & \text{good\_hotel(ritz)} & \leftarrow @\text{aver}(\&\text{godel}(0.8,1), \\ & & @\text{very}(\text{close(ritz,metro)})). \end{array} \right.$$
  
$$\Pi_6 = \left\{ \begin{array}{lll} R_1 : & \text{vanguardist(hydropolis)} & \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} & \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} & \leftarrow 0.7. \\ R_{4-1-3} : & \text{good\_hotel(hydropolis)} & \leftarrow 0.38. \\ R_{4-2} : & \text{good\_hotel(ritz)} & \leftarrow @\text{aver}(\&\text{godel}(0.8,1), \\ & & @\text{very}(\text{close(ritz,metro)})). \end{array} \right.$$

## Ejemplo ( $\Pi_7$ )

$$\Pi_6 = \left\{ \begin{array}{lll} R_1 : & \text{vanguardist(hydropolis)} & \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} & \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} & \leftarrow 0.7. \\ R_{4-1-3} : & \text{good\_hotel(hydropolis)} & \leftarrow 0.38. \\ R_{4-2} : & \text{good\_hotel(ritz)} & \leftarrow @_{\text{aver}}(\&_{\text{godel}}(0.8,1), \\ & & @_{\text{very}}(\text{close(ritz,metro)})). \end{array} \right.$$

$$\Pi_7 = \left\{ \begin{array}{lll} R_1 : & \text{vanguardist(hydropolis)} & \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} & \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} & \leftarrow 0.7. \\ R_{4-1-3} : & \text{good\_hotel(hydropolis)} & \leftarrow 0.38. \\ R_{4-2F} : & \text{good\_hotel(ritz)} & \leftarrow @_{\text{aver}}(\&_{\text{godel}}(0.8,1), \\ & & @_{\text{very}}(0)). \end{array} \right.$$

## Ejemplo ( $\Pi_{10}$ )

$$\Pi_7 = \left\{ \begin{array}{lll} R_1 : & \text{vanguardist(hydropolis)} & \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} & \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} & \leftarrow 0.7. \\ R_{4-1-3////} : & \text{good\_hotel(hydropolis)} & \leftarrow 0.38. \\ R_{4-2F} : & \text{good\_hotel(ritz)} & \leftarrow @aver (&godel(0.8,1), \\ & & @very(0)). \end{array} \right.$$

$$\Pi_{10} = \left\{ \begin{array}{lll} R_1 : & \text{vanguardist(hydropolis)} & \leftarrow 0.9. \\ R_2 : & \text{elegant(ritz)} & \leftarrow 0.8. \\ R_3 : & \text{close(hydropolis,taxi)} & \leftarrow 0.7. \\ R_{4-1-3////} : & \text{good\_hotel(hydropolis)} & \leftarrow 0.38. \\ R_{4-2FIII} : & \text{good\_hotel(ritz)} & \leftarrow 0.4. \end{array} \right.$$

# Despliegado en el entorno FLOPER online

FLOPER

[Documentation](#) [Downloads](#) [Try it online](#)

## Try it!

### Program

```
vanguardist(hydropolis) <- 0.9.  
elegant(ritz) <- 0.8.  
close(hydropolis, taxi) <- 0.7.  
good_hotel(X) <- @aver(elegant(X), @very(close(X, metro))).
```

[symbolic.fpl](#) [rits.fpl](#) [good\\_hotel.fpl](#)

[\*] Unfold program (0) | « Previous | Next »

# Despliegado en el entorno FLOPER online

FLOPER

[Documentation](#) [Downloads](#) [Try it online](#)

## Try it!

### Program

```
vanguardist(hydropolis) <- 0.9.  
elegant(ritz) <- 0.8.  
close(hydropolis, taxi) <- 0.7.  
good_hotel(X) <- @aver(elegant(X), @very(close(X, metro))).
```

[symbolic.fpl](#) [lists.fpl](#) [good\\_hotel.fpl](#)

[ ] Unfold program (0) | « Previous | Next »

Select the rule to unfold:

- 1 vanguardist(hydropolis) <- 0.9.
- 2 elegant(ritz) <- 0.8.
- 3 close(hydropolis, taxi) <- 0.7.
- 4 good\_hotel(X) <- @aver(elegant(X), @very(close(X, metro))).

# Despliegado en el entorno FLOPER online

FLOPER

[Documentation](#) [Downloads](#) [Try it online](#)

## Try it!

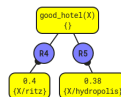
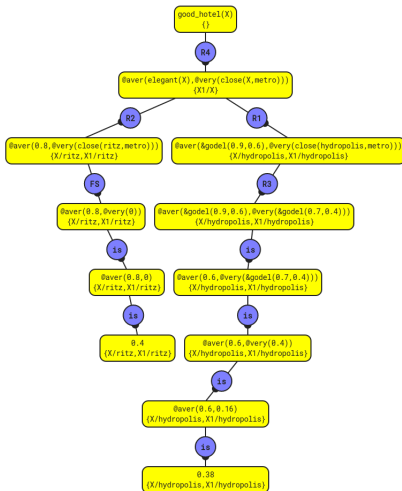
### Program

```
vanguardist(hydropolis) <- 0.9.  
elegant(ritz) <- 0.8.  
close(hydropolis, taxi) <- 0.7.  
good_hotel(ritz) <- @aver(0.8, @very(close(ritz, metro))).  
good_hotel(hydropolis) <- @aver((0.9 &godel 0.6), @very(close(hydropolis, metro))).
```

[symbolic.fpl](#) [lists.fpl](#) [good\\_hotel.fpl](#)

[+] Unfold program (1) | « Previous | Next »

# Despliegado en el entorno FLOPER online



- 1 Motivación y Objetivos
- 2 Lenguaje FASILL
  - Sintaxis
  - Semántica operacional
  - El entorno FLOPER online
- 3 Despliegado de programas difusos
  - Definición
  - Ejemplo
  - Despliegado en el entorno FLOPER online
- 4 Lenguaje sFASILL
  - Sintaxis
  - Semántica operacional
  - Despliegado de programas simbólicos
- 5 Calibrado de programas simbólicos
  - Definición
  - Ejemplo
  - Calibrado en el entorno FLOPER online
- 6 Conclusiones y trabajo futuro

## Lenguaje sFASILL

### sFASILL

sFASILL (*symbolic Fuzzy Aggregators and Similarity Into a Logic Language*) es una extensión de FASILL que permite incluir **valores simbólicos** y **conectivas simbólicas** que no pertenecen a  $L$ .

Un **programa sFASILL**  $\mathcal{P}^s = \langle \Pi^s, \mathcal{R}, L \rangle$  contiene un conjunto  $\Pi^s$  de reglas FASILL simbólicas (o reglas sFASILL).

## Conectivas simbólicas

#& <sub>label</sub>	Conjunción simbólica
#  <sub>label</sub>	Disyunción simbólica
#@ <sub>label</sub>	Agregador simbólico
#? <sub>label</sub>	Conectiva simbólica
# <sub>label</sub>	Grado de verdad simbólico

## Reglas simbólicas

### Regla sFASILL

Una **regla sFASILL** es de la forma  $A \leftarrow B \in \Pi^s$ , donde  $A$  (la cabeza) es una fórmula atómica y  $B$  (el cuerpo) es una fórmula bien formada, construida a partir de fórmulas atómicas  $B_1, B_2, \dots, B_n$ , grados de verdad de  $L$ , valores simbólicos y conectivas –posiblemente simbólicas–.

---

$R_1$  : vanguardist(hydropolis)  $\leftarrow \#_{s1}$ .  
 $R_2$  : elegant(ritz)  $\leftarrow \#_{s2}$ .  
 $R_3$  : close(hydropolis,taxi)  $\leftarrow 0.7$ .  
 $R_4$  : good\_hotel(X)  $\leftarrow \#_{@s3}(\text{elegant}(X),$   
 $\quad \text{@}_{\text{very}}(\text{close}(X,\text{metro}))).$

## Semántica operacional

Ahora los **pasos interpretativos**  $\rightsquigarrow$  **IS** pueden devolver grados de verdad o expresiones con conectivas simbólicas, que deben ser reemplazadas antes de poder computarse su valor.

$\langle \text{good\_hotel}(X); id \rangle$

SS  
 $\rightsquigarrow$

$\langle \#_{s3}(\text{elegant}(\text{ritz}), @_{\text{very}}(\text{close}(\text{ritz}, \text{metro}))); \{X/\text{ritz}\} \rangle$

SS  
 $\rightsquigarrow$

$\langle \#_{s3}(\&_{\text{godel}}(1, \#_{s2}), @_{\text{very}}(\text{close}(\text{ritz}, \text{metro}))); \{X/\text{ritz}\} \rangle$

FS  
 $\rightsquigarrow$

$\langle \#_{s3}(\&_{\text{godel}}(1, \#_{s2}), @_{\text{very}}(0)); \{X/\text{ritz}\} \rangle$

IS  
 $\rightsquigarrow$

$\langle \#_{s3}(\&_{\text{godel}}(1, \#_{s2}), 0); \{X/\text{ritz}\} \rangle$

# Despliegado de programas simbólicos

FLOPER

[Documentation](#) [Downloads](#) [Try it online](#)

## Try it!

### Program

```
vanguardist(hydropolis) <- #s1.  
elegant(ritz) <- #s2.  
close(hydropolis, taxi) <- 0.7.  
good_hotel(X) <- ##s3(elegant(X), @very(close(X, metro))).
```

[symbolic.fpl](#) [lists.fpl](#) [good\\_hotel.fpl](#)

[\[-\] Unfold program \(0\) | « Previous | Next »](#)

Select the rule to unfold:

```
1 vanguardist(hydropolis) <- #s1.  
2 elegant(ritz) <- #s2.  
3 close(hydropolis, taxi) <- 0.7.  
4 good_hotel(X) <- ##s3(elegant(X), @very(close(X, metro))).
```

# Despliegado de programas simbólicos

FLOPER

[Documentation](#) [Downloads](#) [Try it online](#)

## Try it!

### Program

```
vanguardist(hydropolis) <- #s1.  
elegant(ritz) <- #s2.  
close(hydropolis, taxi) <- 0.7.  
good_hotel(ritz) <- #s3(#s2, 0).  
good_hotel(hydropolis) <- #s3((#s1 &godel 0.6), 0.16).
```

[symbolic.fpl](#) [lists.fpl](#) [good\\_hotel.fpl](#)

[\*] Unfold program (6) | « Previous | Next »

- 1 Motivación y Objetivos
- 2 Lenguaje FASILL
  - Sintaxis
  - Semántica operacional
  - El entorno FLOPER online
- 3 Despliegado de programas difusos
  - Definición
  - Ejemplo
  - Despliegado en el entorno FLOPER online
- 4 Lenguaje sFASILL
  - Sintaxis
  - Semántica operacional
  - Despliegado de programas simbólicos
- 5 Calibrado de programas simbólicos
  - Definición
  - Ejemplo
  - Calibrado en el entorno FLOPER online
- 6 Conclusiones y trabajo futuro

# Sustituciones simbólicas

## Sustitución simbólica

Una **sustitución simbólica**  $\Theta$  es una aplicación que asocia a valores y conectivas simbólicas expresiones sobre el conjunto de constantes y conectivas de  $L$ .

$$\Theta = \{ \#_{s1}/0.3, \#_{s2}/0.5, \#_{@s3}/@_{aver} \}$$

## Casos de prueba

### Caso de prueba

Un **caso de prueba** para un programa  $\mathcal{P}^s$  es un par  $(Q, f)$ , donde  $Q$  es un objetivo y  $f$  es una respuesta computada difusa para  $Q$  en  $\mathcal{P}^s$ .

---

$(\text{good\_hotel}(\text{hydropolis}), \langle 0.35; id \rangle)$   
 $(\text{good\_hotel}(\text{ritz}), \langle 0.1; id \rangle)$

# Calibrado de programas simbólicos

## Calibrado de programas simbólicos

El **calibrado de programas simbólicos** consiste en buscar la **sustitución simbólica** que minimiza el error frente a un conjunto de **casos de prueba** esperados.

El objetivo del calibrado es encontrar de manera eficiente los grados de verdad y las conectivas adecuadas para las reglas de un programa.

- Calibrado básico
- Calibrado simbólico
- Calibrado umbralizado

## Ejemplo

$$\mathcal{P}^s = \begin{cases} R_1 : & \text{vanguardist(hydropolis)} & \leftarrow \#_{s1}. \\ R_2 : & \text{elegant(ritz)} & \leftarrow \#_{s2}. \\ R_3 : & \text{close(hydropolis,taxi)} & \leftarrow 0.7. \\ R_4 : & \text{good\_hotel}(X) & \leftarrow \#_{@s3}(\text{elegant}(X), \\ & & \text{@}_{\text{very}}(\text{close}(X,\text{metro})). \end{cases}$$

$$\Sigma_L^T = \{0.3, 0.5, 0.7\}$$

$$\Sigma_L^C = \{\text{@}_{\text{aver}}, \text{@}_{\text{geom}}\}$$

$$t = \begin{cases} (\text{good\_hotel}(\text{hydropolis}), \langle 0.35; id \rangle) \\ (\text{good\_hotel}(\text{ritz}), \langle 0.1; id \rangle) \end{cases}$$

## Ejemplo (sustituciones simbólicas)

$$\Theta_1 = \{\#_{s1}/0.3, \#_{s2}/0.3, \#_{s3}/@aver\}$$

$$\Theta_2 = \{\#_{s1}/0.3, \#_{s2}/0.5, \#_{s3}/@aver\}$$

$$\Theta_3 = \{\#_{s1}/0.3, \#_{s2}/0.7, \#_{s3}/@aver\}$$

$$\Theta_4 = \{\#_{s1}/0.5, \#_{s2}/0.3, \#_{s3}/@aver\}$$

$$\Theta_5 = \{\#_{s1}/0.5, \#_{s2}/0.5, \#_{s3}/@aver\}$$

$$\Theta_6 = \{\#_{s1}/0.5, \#_{s2}/0.7, \#_{s3}/@aver\}$$

$$\Theta_7 = \{\#_{s1}/0.7, \#_{s2}/0.3, \#_{s3}/@aver\}$$

$$\Theta_8 = \{\#_{s1}/0.7, \#_{s2}/0.5, \#_{s3}/@aver\}$$

$$\Theta_9 = \{\#_{s1}/0.7, \#_{s2}/0.7, \#_{s3}/@aver\}$$

$$\Theta_{10} = \{\#_{s1}/0.3, \#_{s2}/0.3, \#_{s3}/@geom\}$$

$$\Theta_{11} = \{\#_{s1}/0.3, \#_{s2}/0.5, \#_{s3}/@geom\}$$

$$\Theta_{12} = \{\#_{s1}/0.3, \#_{s2}/0.7, \#_{s3}/@geom\}$$

$$\Theta_{13} = \{\#_{s1}/0.5, \#_{s2}/0.3, \#_{s3}/@geom\}$$

$$\Theta_{14} = \{\#_{s1}/0.5, \#_{s2}/0.5, \#_{s3}/@geom\}$$

$$\Theta_{15} = \{\#_{s1}/0.5, \#_{s2}/0.7, \#_{s3}/@geom\}$$

$$\Theta_{16} = \{\#_{s1}/0.7, \#_{s2}/0.3, \#_{s3}/@geom\}$$

$$\Theta_{17} = \{\#_{s1}/0.7, \#_{s2}/0.5, \#_{s3}/@geom\}$$

$$\Theta_{18} = \{\#_{s1}/0.7, \#_{s2}/0.7, \#_{s3}/@geom\}$$

## Ejemplo (calibrado básico)

$$\mathcal{P}^s \Theta_1 = \begin{cases} R_1 : & \text{vanguardist(hydropolis)} & \leftarrow 0.3. \\ R_2 : & \text{elegant(ritz)} & \leftarrow 0.3. \\ R_3 : & \text{close(hydropolis,taxi)} & \leftarrow 0.7. \\ R_4 : & \text{good\_hotel(X)} & \leftarrow @_{\text{aver}}(\text{elegant(X)}, \\ & & @_{\text{very}}(\text{close(X,metro)})). \end{cases}$$

$$D_1 : \langle \text{good\_hotel(ritz); id} \rangle \begin{matrix} \text{SS} \\ \rightsquigarrow \\ \text{SS} \\ \rightsquigarrow \\ \text{FS} \\ \rightsquigarrow \\ \text{IS} \\ \rightsquigarrow^* \end{matrix}$$

$$\langle @_{\text{very}}(\text{elegant(ritz)}, @_{\text{very}}(\text{close(ritz,metro)})); id \rangle$$

$$\langle @_{\text{very}}(\&_{\text{godel}}(1,0.3), @_{\text{very}}(\text{close(ritz,metro)})); id \rangle$$

$$\langle @_{\text{very}}(\&_{\text{godel}}(1,0.3), @_{\text{very}}(0)); id \rangle$$

$$\langle 0.15; id \rangle$$

$$D_2 : \langle \text{good\_hotel(hydropolis); id} \rangle \rightsquigarrow^* \langle 0.23; id \rangle$$

## Ejemplo (calibrado básico)

#@ <sub>s3</sub>	# <sub>s1</sub>	# <sub>s2</sub>	$\Theta$	ritz		hydropolis		z
@ <sub>aver</sub>	0.3	0.3	$\Theta_1$	0.15	0.05	0.23	0.12	0.17
		0.5	$\Theta_2$	0.25	0.15	0.23	0.12	0.27
		0.7	$\Theta_3$	0.35	0.25	0.23	0.12	0.37
	0.5	0.3	$\Theta_4$	0.15	0.05	0.33	0.02	0.07
		0.5	$\Theta_5$	0.25	0.15	0.33	0.02	0.17
		0.7	$\Theta_6$	0.35	0.25	0.33	0.02	0.27
	0.7	0.3	$\Theta_7$	0.15	0.05	0.38	0.03	0.08
		0.5	$\Theta_8$	0.25	0.15	0.38	0.05	0.2
		0.7	$\Theta_9$	0.35	0.25	0.38	0.05	0.3
@ <sub>geom</sub>	0.3	0.3	$\Theta_{10}$	0	0.1	0.21	0.14	0.24
		0.5	$\Theta_{11}$	0	0.1	0.21	0.14	0.24
		0.7	$\Theta_{12}$	0	0.1	0.21	0.14	0.24
	0.5	0.3	$\Theta_{13}$	0	0.1	0.28	0.07	0.17
		0.5	$\Theta_{14}$	0	0.1	0.28	0.07	0.17
		0.7	$\Theta_{15}$	0	0.1	0.28	0.07	0.17
	0.7	0.3	$\Theta_{16}$	0	0.1	0.31	0.04	0.14
		0.5	$\Theta_{17}$	0	0.1	0.31	0.04	0.14
		0.7	$\Theta_{18}$	0	0.1	0.31	0.04	0.14

## Ejemplo (calibrado simbólico)

$D_1$  : `<good_hotel(ritz); id>`  
`<#@s3(elegant(ritz),@very(close(ritz,metro))); id>`  
`<#@s3(&godel(1,#s2),@very(close(ritz,metro))); id>`  
`<#@s3(&godel(1,#s2),@very(0)); id>`  
`<#@s3(&godel(1,#s2),0); id>`

SS  
~  
SS  
~  
FS  
~  
IS  
~

$D_2$  : `<good_hotel(hydropolis); id>`  
`<#@s3(elegant(hydropolis),@very(close(hydropolis,metro))); id>`  
`<#@s3(&godel(0.6,#s1),@very(close(hydropolis,metro))); id>`  
`<#@s3(&godel(0.6,#s1),@very(&godel(0.4,0.7))); id>`  
`<#@s3(&godel(0.6,#s1),@very(0.4)); id>`  
`<#@s3(&godel(0.6,#s1),0.16); id>`

SS  
~  
SS  
~  
SS  
~  
IS  
~  
IS  
~

## Ejemplo (calibrado umbralizado)

#@ <sub>s3</sub>	# <sub>s1</sub>	# <sub>s2</sub>	$\Theta$	ritz		hydropolis		z
@ <sub>aver</sub>	0.3	0.3	$\Theta_1$	0.15	0.05	0.23	0.12	<b>0.17</b>
		0.5	$\Theta_2$	0.25	0.15	0.23	0.12	0.27
		0.7	$\Theta_3$	0.35	<b>0.25</b>	<b>0.23</b>	<b>0.12</b>	<b>0.37</b>
	0.5	0.3	$\Theta_4$	0.15	<b>0.05</b>	<b>0.33</b>	<b>0.02</b>	<b>0.07</b>
		0.5	$\Theta_5$	0.25	<b>0.15</b>	<b>0.33</b>	<b>0.02</b>	<b>0.17</b>
		0.7	$\Theta_6$	0.35	<b>0.25</b>	<b>0.33</b>	<b>0.02</b>	<b>0.27</b>
	0.7	0.3	$\Theta_7$	0.15	0.05	0.38	0.03	0.08
		0.5	$\Theta_8$	0.25	<b>0.15</b>	<b>0.38</b>	<b>0.05</b>	<b>0.2</b>
		0.7	$\Theta_9$	0.35	<b>0.25</b>	<b>0.38</b>	<b>0.05</b>	<b>0.3</b>
@ <sub>geom</sub>	0.3	0.3	$\Theta_{10}$	0	<b>0.1</b>	<b>0.21</b>	<b>0.14</b>	<b>0.24</b>
		0.5	$\Theta_{11}$	0	<b>0.1</b>	<b>0.21</b>	<b>0.14</b>	<b>0.24</b>
		0.7	$\Theta_{12}$	0	<b>0.1</b>	<b>0.21</b>	<b>0.14</b>	<b>0.24</b>
	0.5	0.3	$\Theta_{13}$	0	<b>0.1</b>	<b>0.28</b>	<b>0.07</b>	<b>0.17</b>
		0.5	$\Theta_{14}$	0	<b>0.1</b>	<b>0.28</b>	<b>0.07</b>	<b>0.17</b>
		0.7	$\Theta_{15}$	0	<b>0.1</b>	<b>0.28</b>	<b>0.07</b>	<b>0.17</b>
	0.7	0.3	$\Theta_{16}$	0	<b>0.1</b>	<b>0.31</b>	<b>0.04</b>	<b>0.14</b>
		0.5	$\Theta_{17}$	0	<b>0.1</b>	<b>0.31</b>	<b>0.04</b>	<b>0.14</b>
		0.7	$\Theta_{18}$	0	<b>0.1</b>	<b>0.31</b>	<b>0.04</b>	<b>0.14</b>



# Calibrado en el entorno FLOPER online

Run Tune

Test cases

```
0.35 -> good_hotel(ritz).  
0.1 -> good_hotel(hydropolis).
```

symbolic test

Basic

```
< 0.13, {#s1/0.3, #s2/0.7, #s3/@aver} >
```

Executed in 10 milliseconds!

Symbolic

```
< 0.13, {#s1/0.3, #s2/0.7, #s3/@aver} >
```

Executed in 10 milliseconds!

Thresholded

```
< 0.13, {#s1/0.3, #s2/0.7, #s3/@aver} >
```

Executed in 0 milliseconds!

Apply substitution Generate substitution

# Calibrado en el entorno FLOPER online

FLOPER

[Documentation](#) [Downloads](#) [Try it online](#)

## Try it!

### Program

```
vanguardist(hydropolis) <- 0.3.  
elegant(ritz) <- 0.7.  
close(hydropolis, taxi) <- 0.7.  
good_hotel(X) <- @aver(elegant(X), @very(close(X, metro))).
```

[symbolic.fpl](#) [futs.fpl](#) [good\\_hotel.fpl](#)

[\*] Unfold program (0) | « Previous | Next »

- 1 Motivación y Objetivos
- 2 Lenguaje FASILL
  - Sintaxis
  - Semántica operacional
  - El entorno FLOPER online
- 3 Despliegado de programas difusos
  - Definición
  - Ejemplo
  - Despliegado en el entorno FLOPER online
- 4 Lenguaje sFASILL
  - Sintaxis
  - Semántica operacional
  - Despliegado de programas simbólicos
- 5 Calibrado de programas simbólicos
  - Definición
  - Ejemplo
  - Calibrado en el entorno FLOPER online
- 6 Conclusiones y trabajo futuro

## Conclusiones

- Ampliación de las capacidades del entorno FLOPER.
  - Despliegado de programas difusos.
  - Extensiones simbólicas de MALP y FASILL.
  - Calibrado automático de programas difusos simbólicos.
- Formalismos subyacentes de la lógica difusa y la programación lógica difusa, así como de las técnicas de despliegado y calibrado simbólico implementadas.

**Publicaciones presentadas en congresos internacionales**  
**LOPSTR'16, RuleML+RR'17, SUM'17 y ESCIM'17**

## Contribuciones esenciales

- 1 Implementación del despliegado difuso. Integración en el entorno online.
- 2 Implementación de mecanismos para gestionar versiones de programas.
- 3 Definición de la sintaxis de los elementos simbólicos.
- 4 Actualización del análisis para las extensiones simbólicas.
- 5 Actualización de la semántica operacional para extensiones simbólicas.
- 6 Definición de la sintaxis de los casos de prueba.
- 7 Implementación del análisis de los casos de prueba.
- 8 Implementación de métodos de calibrado. Integración en el entorno online.
- 9 Medición del tiempo de ejecución de los métodos de calibrado.

## Trabajo futuro

- **Corrección del despliegado**

El despliegado FASILL no siempre mantiene la semántica del programa original cuando intervienen pasos  $\overset{\text{FS}}{\rightsquigarrow}$ .

$$\mathcal{R}(p, q) = 0.8$$

$$\mathcal{P} = \begin{cases} R_1 : p(X) \leftarrow @_{\text{aver}}(q(X), 1). \\ R_2 : r(a) \leftarrow 0.6. \end{cases}$$

$$\langle p(b); id \rangle \rightsquigarrow^* \langle 0.5; \{X/b\} \rangle$$

$$\mathcal{P}' = \begin{cases} R_{1-2} : p(a) \leftarrow @_{\text{aver}}(\&_{\text{godel}}(0.8, 0.6), 1). \\ R_2 : r(a) \leftarrow 0.6. \end{cases}$$

$$\langle p(b); id \rangle \rightsquigarrow^* \langle 0; id \rangle$$

## Trabajo futuro

- **Corrección del despliegado**

El despliegado FASILL no siempre mantiene la semántica del programa original debido a las propiedades de las conectivas.

$$\mathcal{R}(a, b) = 0.4, \quad \mathcal{R}(q, r) = 0.5$$

$$\mathcal{P} = \begin{cases} R_1 : p(X) \leftarrow @_{\text{very}}(q(X)). \\ R_2 : r(b) \leftarrow 1. \end{cases}$$

$$\langle p(a); id \rangle \rightsquigarrow^* \langle 0.16; \{X/a\} \rangle$$

$$\mathcal{P}' = \begin{cases} R_{1-2} : p(b) \leftarrow @_{\text{very}}(\&_{\text{godel}}(0.5, 1)). \\ R_2 : r(b) \leftarrow 1. \end{cases}$$

$$\langle p(a); id \rangle \rightsquigarrow^* \langle 0.25; id \rangle$$

## Trabajo futuro

- **Calibrado de programas simbólicos**

sFASILL solo permite introducir grados de verdad y conectivas simbólicas en las reglas del programa.

En el futuro, nos proponemos permitir que las relaciones de similitud acepten también elementos simbólicos.

$$\mathcal{R}(\text{elegant}, \text{vanguardist}) = \#_{s1}$$
$$\text{t-norm} = \#_{s2}$$

Fin