Building a Fuzzy Transformation System

Ginés Moreno

Dep. Computer Science // U. Castilla – La Mancha // Spain

Outline of the talk

- Introduction and Aim of the Work.
- Multi-Adjoint Logic Programs.
- Fuzzy Transformation Rules.
- Conclusions and Further Research.

Introduction and Aim of the Work

- The problem: Optimizing fuzzy logic programs by means of Fold/Unfold transformations.

Introduction and Aim of the Work

- The problem: Optimizing fuzzy logic programs by means of Fold/Unfold transformations.
- Starting point: An extremely flexible fuzzy logic language and our experience on previous functional–logic and fuzzy–logic transformations.

Introduction and Aim of the Work

- The problem: Optimizing fuzzy logic programs by means of Fold/Unfold transformations.
- Starting point: An extremely flexible fuzzy logic language and our experience on previous functional–logic and fuzzy–logic transformations.
- Developed work: A complete set of fuzzy transformation rules for multi–adjoint logic programs.

Introduction and Aim of the Work

- The problem: Optimizing fuzzy logic programs by means of Fold/Unfold transformations.
- Starting point: An extremely flexible fuzzy logic language and our experience on previous functional–logic and fuzzy–logic transformations.
- Developed work: A complete set of fuzzy transformation rules for multi–adjoint logic programs.
- Results: Strong correctness of the transformation system and gains in efficiency on final programs.

Introduction and Aim of the Work

PROGRAM TRANSFORMATION BY FOLD/UNFOLD

From $\mathcal{P}_0$ derive a sequence $\mathcal{P}_1, \ldots, \mathcal{P}_n$, such that:

Introduction and Aim of the Work

PROGRAM TRANSFORMATION BY FOLD/UNFOLD

From $\mathcal{P}_0$ derive a sequence $\mathcal{P}_1, \ldots, \mathcal{P}_n$, such that:

- $\mathcal{P}_i$ is obtained from $\mathcal{P}_{i-1}$ by folding, unfolding, etc...

Introduction and Aim of the Work

PROGRAM TRANSFORMATION BY FOLD/UNFOLD

From $\mathcal{P}_0$ derive a sequence $\mathcal{P}_1, \ldots, \mathcal{P}_n$, such that:

- $\mathcal{P}_i$ is obtained from $\mathcal{P}_{i-1}$ by folding, unfolding, etc...
- Each $\mathcal{P}_i$ produces the same outputs than $\mathcal{P}_0$.

Introduction and Aim of the Work

PROGRAM TRANSFORMATION BY FOLD/UNFOLD

From $\mathcal{P}_0$ derive a sequence $\mathcal{P}_1, \ldots, \mathcal{P}_n$, such that:

- $\mathcal{P}_i$ is obtained from $\mathcal{P}_{i-1}$ by folding, unfolding, etc...
- Each $\mathcal{P}_i$ produces the same outputs than $\mathcal{P}_0$.
- $\mathcal{P}_n$ "is better" (i.e., it runs faster) than $\mathcal{P}_0$.

Introduction and Aim of the Work

FUZZY LOGIC PROGRAMMING

Introduction and Aim of the Work

FUZZY LOGIC PROGRAMMING

$$\Downarrow$$

FUZZY LOGIC

+

LOGIC PROGRAMMING

Introduction and Aim of the Work

- Although there is no an standard language, we have found two major approaches:

Introduction and Aim of the Work

- Although there is no an standard language, we have found two major approaches:

1. LIKELOG *[Arcelli & Formato-99]*

SLD-resolution + **FUZZY (similarity)** unification

Introduction and Aim of the Work

- Although there is no an standard language, we have found two major approaches:

1. LIKELOG *[Arcelli & Formato-99]*

SLD-resolution + <u>**FUZZY (similarity)** unification</u>

2. f-Prolog *[Vojtas & Paulík-96]*

<u>**FUZZY** SLD-resolution</u> + (syntactic) unification

Introduction and Aim of the Work

- Although there is no an standard language, we have found two major approaches:

1. LIKELOG *[Arcelli & Formato-99]*

SLD-resolution + **FUZZY (similarity) unification**

2. f-Prolog *[Vojtas & Paulík-96]*

FUZZY SLD-resolution + (syntactic) unification

- Multi–adjoint *[Medina & Ojeda–Aciego & Vojtas-01]*

Admissible/Interpretive Computation + (syntactic) unification

Multi-Adjoint Logic Programs

- Let $\mathcal{L}$ be a first order language containing:

constants	variables	functions
predicates	quantifiers: $\forall, \exists$	connectives:

$$\begin{array}{lllll} \&_1, & \&_2, & \ldots, & \&_k & \texttt{(conjunctions)} \\ \vee_1, & \vee_2, & \ldots, & \vee_l & \texttt{(disjunctions)} \\ \leftarrow_1, & \leftarrow_2, & \ldots, & \leftarrow_m & \texttt{(implications)} \\ @_1, & @_2, & \ldots, & @_n & \texttt{(aggregations)} \end{array}$$

Multi-Adjoint Logic Programs

- Let $\mathcal{L}$ be a first order language containing:

  constants, variables, functions,
  predicates, quantifiers: $\forall, \exists$, connectives:

  $$\&_1, \quad \&_2, \quad \ldots, \quad \&_k \qquad \texttt{(conjunctions)}$$
  $$\vee_1, \quad \vee_2, \quad \ldots, \quad \vee_l \qquad \texttt{(disjunctions)}$$
  $$\leftarrow_1, \quad \leftarrow_2, \quad \ldots, \quad \leftarrow_m \qquad \texttt{(implications)}$$
  $$@_1, \quad @_2, \quad \ldots, \quad @_n \qquad \texttt{(aggregations)}$$

- $\mathcal{L}$ also contains values $r \in L$ of a multi–adjoint lattice, $\langle L, \preceq, \leftarrow_1, \&_1, \ldots, \leftarrow_n, \&_n \rangle$. For instance, $\langle [\mathbf{0}, \mathbf{1}], \preceq, \leftarrow_{\mathbf{luka}}, \&_{\mathbf{luka}}, \leftarrow_{\mathbf{prod}}, \&_{\mathbf{prod}}, \leftarrow_{\mathbf{G}}, \&_{\mathbf{G}} \rangle$

Multi-Adjoint Logic Programs

- SYNTAX: A program rule is $\mathtt{A} \ \leftarrow_{\mathtt{i}} \ \mathcal{B} \ \textbf{with} \ \alpha$ where $\alpha \in L$ is the truth degree of the rule

Multi-Adjoint Logic Programs

- SYNTAX: A program rule is $\mathtt{A} \leftarrow_{\mathtt{i}} \mathcal{B}$ **with** α where $\alpha \in L$ is the truth degree of the rule

$$\begin{array}{llll}
\mathcal{R}_1 : p(X) & \leftarrow_{\mathtt{prod}} & q(X,Y) \&_{\mathtt{G}} r(Y); & \text{with} \quad 0.8 \\
\mathcal{R}_2 : q(a,Y) & \leftarrow_{\mathtt{prod}} & s(Y); & \text{with} \quad 0.7 \\
\mathcal{R}_3 : q(Y,a) & \leftarrow_{\mathtt{luka}} & r(Y); & \text{with} \quad 0.8 \\
\mathcal{R}_4 : r(Y) & \leftarrow_{\mathtt{luka}} ; & & \text{with} \quad 0.6 \\
\mathcal{R}_5 : s(b) & \leftarrow_{\mathtt{luka}} ; & & \text{with} \quad 0.9
\end{array}$$

Multi-Adjoint Logic Programs

- SYNTAX: A program rule is $\mathtt{A} \leftarrow_{\mathtt{i}} \mathcal{B}$ **with** α where $\alpha \in L$ is the truth degree of the rule

$$\begin{array}{llll}
\mathcal{R}_1 : p(X) & \leftarrow_{\mathtt{prod}} & q(X,Y) \&_{\mathtt{G}} r(Y); & \text{with} \quad 0.8 \\
\mathcal{R}_2 : q(a,Y) & \leftarrow_{\mathtt{prod}} & s(Y); & \text{with} \quad 0.7 \\
\mathcal{R}_3 : q(Y,a) & \leftarrow_{\mathtt{luka}} & r(Y); & \text{with} \quad 0.8 \\
\mathcal{R}_4 : r(Y) & \leftarrow_{\mathtt{luka}} ; & & \text{with} \quad 0.6 \\
\mathcal{R}_5 : s(b) & \leftarrow_{\mathtt{luka}} ; & & \text{with} \quad 0.9
\end{array}$$

- INPUT (goal): Expression similar to the body of a program rule. For instance, $\leftarrow p(X) \;\&_{\mathtt{G}}\; r(a)$

Multi-Adjoint Logic Programs

- STATE : Is a pair with form $\boxed{\langle goal;\ substitution \rangle}$

Multi-Adjoint Logic Programs

- STATE : Is a pair with form $\boxed{\langle goal;\ substitution\rangle}$
- OUTPUT (fuzzy computed answer): Is a (final) state of the form $\boxed{\langle truth\_degree;\ substitution\rangle}$

Multi-Adjoint Logic Programs

- STATE : Is a pair with form $\boxed{\langle goal;\ substitution\rangle}$
- OUTPUT (fuzzy computed answer): Is a (final) state of the form $\boxed{\langle truth\_degree;\ substitution\rangle}$
- PROCEDURAL SEMANTICS:
  Operational phase: Admissible steps ($\rightarrow_{AS}$)
  Interpretive phase: Interpretive steps ($\rightarrow_{IS}$)

Multi-Adjoint Logic Programs

- STATE : Is a pair with form $\boxed{\langle goal;\ substitution \rangle}$
- OUTPUT (fuzzy computed answer): Is a (final) state of the form $\boxed{\langle truth\_degree;\ substitution \rangle}$
- PROCEDURAL SEMANTICS:
  Operational phase: Admissible steps ($\to_{AS}$)
  Interpretive phase: Interpretive steps ($\to_{IS}$)
- Given a program $\mathcal{P}$, goal $\mathcal{Q}$ and substitution σ, we define an STATE TRANSITION SYSTEM whose transition relations are $\to_{AS}$ and $\to_{IS}$

Multi-Adjoint Logic Programs

Multi-Adjoint Logic Programs

ADMISSIBLE STEP OF KIND $\rightarrow_{AS1}$

$$\langle \mathcal{Q}[A]; \sigma \rangle \;\; \rightarrow_{AS1} \;\; \langle (\mathcal{Q}[A/v \&_i \mathcal{B}])\theta; \sigma\theta \rangle \text{ if}$$

(1) $\quad A$ is the selected atom in $\mathcal{Q}$,

(2) $\quad \theta = mgu(\{A' = A\})$,

(3) $\quad A' \leftarrow_i \mathcal{B}$ with v in $\mathcal{P}$ and $\mathcal{B}$ is not empty.

Multi-Adjoint Logic Programs

ADMISSIBLE STEP OF KIND $\rightarrow_{AS1}$

$$\langle \mathcal{Q}[A]; \sigma \rangle \ \rightarrow_{AS1} \ \langle (\mathcal{Q}[A/v \&_i \mathcal{B}])\theta; \sigma\theta \rangle \text{ if}$$

(1) A is the selected atom in $\mathcal{Q}$,

(2) $\theta = mgu(\{A' = A\})$,

(3) $A' \leftarrow_i \mathcal{B}$ with v in $\mathcal{P}$ and $\mathcal{B}$ is not empty.

EXAMPLE. Let $\mathtt{p(a)} \leftarrow_{\mathtt{prod}} \mathtt{p(f(a))}$ `with` 0.7 be a rule

$$\langle (\mathtt{p(b)} \&_{\mathtt{G}} \mathtt{p(X)}) \&_{\mathtt{G}} \mathtt{q(X)}; \mathtt{id} \rangle \rightarrow_{AS1}$$
$$\langle (\mathtt{p(b)} \&_{\mathtt{G}} 0.7 \&_{\mathtt{prod}} \mathtt{p(f(a))}) \&_{\mathtt{G}} \mathtt{q(a)}; \{\mathtt{X/a}\} \rangle$$

Multi-Adjoint Logic Programs

ADMISSIBLE STEP OF KIND $\rightarrow_{AS2}$

$\langle \mathcal{Q}[A]; \sigma \rangle \;\; \rightarrow_{AS2} \;\; \langle (\mathcal{Q}[A/v])\theta; \sigma\theta \rangle$ if

(1) A is the selected atom in $\mathcal{Q}$,

(2) $\theta = mgu(\{A' = A\})$, and

(3) $A' \leftarrow_i$ with v in $\mathcal{P}$.

Multi-Adjoint Logic Programs

ADMISSIBLE STEP OF KIND $\rightarrow_{AS2}$

$$\langle \mathcal{Q}[A]; \sigma \rangle \;\; \rightarrow_{AS2} \;\; \langle (\mathcal{Q}[A/v])\theta; \sigma\theta \rangle \text{ if}$$

(1) A is the selected atom in $\mathcal{Q}$,

(2) $\theta = mgu(\{A' = A\})$, and

(3) $A' \leftarrow_i$ with v in $\mathcal{P}$.

EXAMPLE. Let $\mathtt{p(a)} \leftarrow_{\mathtt{luka}}$ $\mathtt{with}$ $\mathtt{0.7}$ be a rule

$$\langle (\mathtt{p(b)} \&_{\mathtt{G}} \mathtt{p(X)}) \&_{\mathtt{G}} \mathtt{q(X)}; \mathtt{id} \rangle \;\; \rightarrow_{AS2}$$
$$\langle (\mathtt{p(b)} \&_{\mathtt{G}} \mathtt{0.7}) \&_{\mathtt{G}} \mathtt{q(a)}; \{\mathtt{X/a}\} \rangle$$

Multi-Adjoint Logic Programs

INTERPRETIVE STEP $\rightarrow_{IS}$

$$\langle Q[@(r_1, r_2)]; \sigma\rangle \;\rightarrow_{IS}\; \langle Q[@(r_1,r_2)/[\![@]\!](r_1,r_2)];\sigma\rangle$$

where $[\![@]\!]$ is the truth function of connective $@$ in the multi-adjoint lattice associated to $\mathcal{P}$

Multi-Adjoint Logic Programs

INTERPRETIVE STEP $\rightarrow_{IS}$

$$\langle Q[@(r_1, r_2)]; \sigma\rangle \;\; \rightarrow_{IS} \;\; \langle Q[@(r_1,r_2)/[\![@]\!](r_1,r_2)];\sigma\rangle$$

where $[\![@]\!]$ is the truth function of connective $@$ in the multi-adjoint lattice associated to $\mathcal{P}$

EXAMPLE. Since the truth function associated to $\&_{\texttt{prod}}$ is the product operator, then

$$\langle (0.8 \&_{\texttt{luka}} ((0.7 \&_{\texttt{prod}} 0.9) \&_{\texttt{G}} 0.7)); \{X/a\}\rangle \;\; \rightarrow_{IS}$$
$$\langle (0.8 \&_{\texttt{luka}} (0.63 \&_{\texttt{G}} 0.7)); \{X/a\}\rangle$$

Building a Fuzzy Transformation System

$\langle \underline{p(X)} \&_{\mathtt{G}} r(a); id \rangle \quad \rightarrow_{AS1}{}^{\mathcal{R}_1}$

$\langle (0.8 \&_{\mathtt{prod}} (\underline{q(X_1, Y_1)} \&_{\mathtt{G}} r(Y_1))) \&_{\mathtt{G}} r(a); \{X/X_1\} \rangle \quad \rightarrow_{AS1}{}^{\mathcal{R}_2}$

$\langle (0.8 \&_{\mathtt{prod}} ((0.7 \&_{\mathtt{prod}} \underline{s(Y_2)}) \&_{\mathtt{G}} r(Y_2))) \&_{\mathtt{G}} r(a); \{X/a, X_1/a, Y_1/Y_2\} \rangle \quad \rightarrow_{AS2}{}^{\mathcal{R}_5}$

$\langle (0.8 \&_{\mathtt{prod}} ((0.7 \&_{\mathtt{prod}} 0.9) \&_{\mathtt{G}} \underline{r(b)})) \&_{\mathtt{G}} r(a); \{X/a, X_1/a, Y_1/b, Y_2/b\} \rangle \quad \rightarrow_{AS2}{}^{\mathcal{R}_4}$

$\langle (0.8 \&_{\mathtt{prod}} ((0.7 \&_{\mathtt{prod}} 0.9) \&_{\mathtt{G}} 0.6)) \&_{\mathtt{G}} \underline{r(a)}; \{X/a, X_1/a, Y_1/b, Y_2/b, Y_3/b\} \rangle \quad \rightarrow_{AS2}{}^{\mathcal{R}_4}$

$\langle (0.8 \&_{\mathtt{prod}} (\underline{(0.7 \&_{\mathtt{prod}} 0.9)} \&_{\mathtt{G}} 0.6)) \&_{\mathtt{G}} 0.6; \{X/a, X_1/a, Y_1/b, Y_2/b, Y_3/b\} \rangle \quad \rightarrow_{IS}$

$\langle (0.8 \&_{\mathtt{prod}} (\underline{0.63 \&_{\mathtt{G}} 0.6})) \&_{\mathtt{G}} 0.6; \{X/a, X_1/a, Y_1/b, Y_2/b, Y_3/b\} \rangle \quad \rightarrow_{IS}$

$\langle (\underline{0.8 \&_{\mathtt{prod}} 0.6}) \&_{\mathtt{G}} 0.6; \{X/a, X_1/a, Y_1/b, Y_2/b, Y_3/b\} \rangle \quad \rightarrow_{IS}$

$\langle \underline{0.48 \&_{\mathtt{G}} 0.6}; \{X/a, X_1/a, Y_1/b, Y_2/b, Y_3/b\} \rangle \quad \rightarrow_{IS}$

$\langle 0.48; \{X/a, X_1/a, Y_1/b, Y_2/b, Y_3/b\} \rangle$

So, the f.c.a (fuzzy computed answer) is $\langle 0.48; \{X/a\} \rangle$

Fuzzy Transformation Rules

- Transformation sequence: $(\mathcal{P}_0, \ldots, \mathcal{P}_k)$, $k \geq 0$.

Fuzzy Transformation Rules

- Transformation sequence: $(\mathcal{P}_0, \ldots, \mathcal{P}_k)$, $k \geq 0$.
- **DEFINITION INTRODUCTION RULE**

$\mathcal{P}_{\mathbf{k+1}} \; = \; \mathcal{P}_{\mathbf{k}} \cup \boxed{\{\ \mathbf{p}(\overline{\mathbf{x_n}}) \leftarrow \mathcal{B} \text{ with } \alpha = \mathbf{1}\ \}}$, where

1) p is *new*, i.e., it does not occur in $\mathcal{P}_0, \ldots, \mathcal{P}_k$
2) $\overline{x_n}$ is the set of variables appearing in $\mathcal{B}$
3) other non-variable symbols in $\mathcal{B}$ belong to $\mathcal{P}_0$

Fuzzy Transformation Rules

- Transformation sequence: $(\mathcal{P}_0, \ldots, \mathcal{P}_k)$, $k \geq 0$.
- **DEFINITION INTRODUCTION RULE**

$\mathcal{P}_{\mathbf{k+1}} = \mathcal{P}_{\mathbf{k}} \cup \boxed{\{\ \mathbf{p}(\overline{\mathbf{x_n}}) \leftarrow \mathcal{B} \text{ with } \alpha = \mathbf{1}\ \}}$, where

1) p is *new*, i.e., it does not occur in $\mathcal{P}_0, \ldots, \mathcal{P}_k$
2) $\overline{x_n}$ is the set of variables appearing in $\mathcal{B}$
3) other non-variable symbols in $\mathcal{B}$ belong to $\mathcal{P}_0$

- **EXAMPLE:** $\mathcal{P}_1 = \mathcal{P}_0 \cup \{\mathcal{R}_6\}$ where the "eureka" rule is $\mathcal{R}_6 : \ \mathtt{new}(\mathtt{X}, \mathtt{Y}) \leftarrow \mathtt{q}(\mathtt{X}, \mathtt{Y}) \ \&_{\mathtt{G}} \ \mathtt{r}(\mathtt{Y})$ with $\alpha = 1$

Fuzzy Transformation Rules

- **FOLDING RULE**

  1) Non-eureka $\mathcal{R} : (A \leftarrow_i \mathcal{B}$ with $\alpha = v) \in \mathcal{P}_k$
  2) Eureka $\mathcal{R}' : (A' \leftarrow \mathcal{B}'$ with $\alpha = 1) \in \mathcal{P}_k$
  3) There exists σ s.t. $\mathcal{B}'\sigma$ is contained in $\mathcal{B}$

$$\mathcal{P}_{\mathbf{k+1}} \;=\; (\mathcal{P}_{\mathbf{k}} - \{\mathcal{R}\}) \cup \boxed{\{\mathbf{A} \leftarrow_{\mathbf{i}} \mathcal{B}[\mathcal{B}'\sigma/\mathbf{A}'\sigma] \text{ with } \alpha = \mathbf{v}\}}$$

Fuzzy Transformation Rules

- **FOLDING RULE**

  1) Non-eureka $\mathcal{R} : (A \leftarrow_i \mathcal{B}$ with $\alpha = v) \in \mathcal{P}_k$
  2) Eureka $\mathcal{R}' : (A' \leftarrow \mathcal{B}'$ with $\alpha = 1) \in \mathcal{P}_k$
  3) There exists σ s.t. $\mathcal{B}'\sigma$ is contained in $\mathcal{B}$

$$\mathcal{P}_{\mathbf{k+1}} = (\mathcal{P}_{\mathbf{k}} - \{\mathcal{R}\}) \cup \boxed{\{\mathbf{A} \leftarrow_{\mathbf{i}} \mathcal{B}[\mathcal{B}'\sigma/\mathbf{A}'\sigma] \text{ with } \alpha = \mathbf{v}\}}$$

- **EXAMPLE:** Folding rule $\mathcal{R}_1$ using eureka $\mathcal{R}_6$

$$\mathcal{R}_1 : \quad \mathtt{p}(\mathtt{X}) \leftarrow \mathtt{q}(\mathtt{X},\mathtt{Y}) \ \&_{\mathtt{G}} \ \mathtt{r}(\mathtt{Y}) \text{ with } \alpha = 0.8$$
$$\mathcal{R}_6 : \ \mathtt{new}(\mathtt{X},\mathtt{Y}) \leftarrow \mathtt{q}(\mathtt{X},\mathtt{Y}) \ \&_{\mathtt{G}} \ \mathtt{r}(\mathtt{Y}) \text{ with } \alpha = 1$$

$$\mathcal{P}_{\mathbf{2}} = (\mathcal{P}_{\mathbf{1}} - \{\mathcal{R}_{\mathbf{1}}\}) \cup \{\mathcal{R}_{\mathbf{7}} : \mathbf{p}(\mathbf{X}) \leftarrow_{\mathtt{prod}} \mathbf{new}(\mathbf{X},\mathbf{Y}) \text{ with } \alpha = \mathbf{0.8}\ \}$$

Fuzzy Transformation Rules

- **UNFOLDING RULE**

Let $\mathcal{R} : (A \leftarrow_i \mathcal{B} \text{ with } \alpha = v) \in \mathcal{P}_k$

$$\mathcal{P}_{\mathbf{k+1}} \;=\; (\mathcal{P}_{\mathbf{k}} - \{\mathcal{R}\}) \cup$$

$$\boxed{\{\mathbf{A}\sigma \leftarrow_{\mathbf{i}} \mathcal{B}' \text{ with } \alpha = \mathbf{v} \mid \langle \mathcal{B}; \mathbf{id} \rangle \rightarrow_{\mathbf{AS/IS}} \langle \mathcal{B}'; \sigma \rangle\}}$$

Fuzzy Transformation Rules

- **UNFOLDING RULE**

  Let $\mathcal{R} : (A \leftarrow_i \mathcal{B} \text{ with } \alpha = v) \in \mathcal{P}_k$

  $$\mathcal{P}_{\mathbf{k+1}} = (\mathcal{P}_{\mathbf{k}} - \{\mathcal{R}\}) \cup$$

  $$\boxed{\{\mathbf{A}\sigma \leftarrow_{\mathbf{i}} \mathcal{B}' \text{ with } \alpha = \mathbf{v} \mid \langle \mathcal{B}; \mathbf{id} \rangle \rightarrow_{\mathbf{AS/IS}} \langle \mathcal{B}'; \sigma \rangle\}}$$

- **EXAMPLE:** To unfold the eureka rule...

  $\mathcal{R}_6 : \ \mathtt{new}(\mathtt{X}, \mathtt{Y}) \leftarrow \mathtt{q}(\mathtt{X}, \mathtt{Y}) \ \&_{\mathtt{G}} \ \mathtt{r}(\mathtt{Y})$ with $\alpha = 1$, then

$$\langle \underline{\mathtt{q}(\mathtt{X}, \mathtt{Y})} \&_{\mathtt{G}} \mathtt{r}(\mathtt{Y}); \mathtt{id} \rangle \rightarrow_{\mathtt{AS1}}{}^{\mathcal{R}_2} \langle (0.7 \&_{\mathtt{prod}} \mathtt{s}(\mathtt{Y}_0)) \&_{\mathtt{G}} \mathtt{r}(\mathtt{Y}_0); \{\mathtt{X}/\mathtt{a}, \mathtt{Y}/\mathtt{Y}_0\} \rangle$$

$$\langle \underline{\mathtt{q}(\mathtt{X}, \mathtt{Y})} \&_{\mathtt{G}} \mathtt{r}(\mathtt{Y}); \mathtt{id} \rangle \rightarrow_{\mathtt{AS1}}{}^{\mathcal{R}_3} \langle (0.8 \&_{\mathtt{luka}} \mathtt{r}(\mathtt{Y}_1)) \&_{\mathtt{G}} \mathtt{r}(\mathtt{a}); \{\mathtt{X}/\mathtt{Y}_1, \mathtt{Y}/\mathtt{a}\} \rangle$$

Fuzzy Transformation Rules

- $\mathcal{P}_{\mathbf{3}} = (\mathcal{P}_{\mathbf{2}} - \{\mathcal{R}_{\mathbf{6}}\}) \cup \{\mathcal{R}_{\mathbf{8}}, \mathcal{R}_{\mathbf{9}}\}$
  $\mathcal{R}_8 : \mathtt{new}(\mathtt{a}, \mathtt{Y}_0) \leftarrow ((0.7 \&_{\mathtt{prod}} \mathtt{s}(\mathtt{Y}_0)) \&_{\mathtt{G}} \mathtt{r}(\mathtt{Y}_0))$ with $\alpha = 1$
  $\mathcal{R}_9 : \mathtt{new}(\mathtt{Y}_1, \mathtt{a}) \leftarrow ((0.8 \&_{\mathtt{luka}} \mathtt{r}(\mathtt{Y}_1)) \&_{\mathtt{G}} \mathtt{r}(\mathtt{a}))$ with $\alpha = 1$

Fuzzy Transformation Rules

- $\mathcal{P}_{\mathbf{3}} = (\mathcal{P}_{\mathbf{2}} - \{\mathcal{R}_{\mathbf{6}}\}) \cup \{\mathcal{R}_{\mathbf{8}}, \mathcal{R}_{\mathbf{9}}\}$
  $\mathcal{R}_8 : \mathtt{new}(\mathtt{a}, \mathtt{Y}_0) \leftarrow ((0.7 \&_{\mathtt{prod}} \mathtt{s}(\mathtt{Y}_0)) \ \&_{\mathtt{G}} \ \mathtt{r}(\mathtt{Y}_0))$ with $\alpha = 1$
  $\mathcal{R}_9 : \mathtt{new}(\mathtt{Y}_1, \mathtt{a}) \leftarrow ((0.8 \ \&_{\mathtt{luka}} \ \mathtt{r}(\mathtt{Y}_1)) \ \&_{\mathtt{G}} \ \mathtt{r}(\mathtt{a}))$ with $\alpha = 1$
- $\mathcal{P}_{\mathbf{4}} = (\mathcal{P}_{\mathbf{3}} - \{\mathcal{R}_{\mathbf{8}}\}) \cup \{\mathcal{R}_{\mathbf{10}}\}$
  $\mathcal{R}_{10} : \mathtt{new}(\mathtt{a}, \mathtt{b}) \leftarrow ((0.7 \&_{\mathtt{prod}} 0.9) \ \&_{\mathtt{G}} \ \mathtt{r}(\mathtt{Y}_0))$ with $\alpha = 1$

Fuzzy Transformation Rules

- $\mathcal{P}_{\mathbf{3}} = (\mathcal{P}_{\mathbf{2}} - \{\mathcal{R}_{\mathbf{6}}\}) \cup \{\mathcal{R}_{\mathbf{8}}, \mathcal{R}_{\mathbf{9}}\}$
  $\mathcal{R}_8 : \mathtt{new}(\mathtt{a}, \mathtt{Y}_0) \leftarrow ((0.7 \&_{\mathtt{prod}} \mathtt{s}(\mathtt{Y}_0)) \ \&_{\mathtt{G}} \ \mathtt{r}(\mathtt{Y}_0))$ with $\alpha = 1$
  $\mathcal{R}_9 : \mathtt{new}(\mathtt{Y}_1, \mathtt{a}) \leftarrow ((0.8 \ \&_{\mathtt{luka}} \ \mathtt{r}(\mathtt{Y}_1)) \ \&_{\mathtt{G}} \ \mathtt{r}(\mathtt{a}))$ with $\alpha = 1$
- $\mathcal{P}_{\mathbf{4}} = (\mathcal{P}_{\mathbf{3}} - \{\mathcal{R}_{\mathbf{8}}\}) \cup \{\mathcal{R}_{\mathbf{10}}\}$
  $\mathcal{R}_{10} : \mathtt{new}(\mathtt{a}, \mathtt{b}) \leftarrow ((0.7 \&_{\mathtt{prod}} 0.9) \ \&_{\mathtt{G}} \ \mathtt{r}(\mathtt{Y}_0))$ with $\alpha = 1$
- $\mathcal{P}_{\mathbf{5}} = (\mathcal{P}_{\mathbf{4}} - \{\mathcal{R}_{\mathbf{10}}\}) \cup \{\mathcal{R}_{\mathbf{11}}\}$
  $\mathcal{R}_{11} : \mathtt{new}(\mathtt{a}, \mathtt{b}) \leftarrow ((0.7 \&_{\mathtt{prod}} 0.9) \ \&_{\mathtt{G}} \ 0.6)$ with $\alpha = 1$

Fuzzy Transformation Rules

- $\mathcal{P}_{\mathbf{3}} = (\mathcal{P}_{\mathbf{2}} - \{\mathcal{R}_{\mathbf{6}}\}) \cup \{\mathcal{R}_{\mathbf{8}}, \mathcal{R}_{\mathbf{9}}\}$
  $\mathcal{R}_8 : \mathtt{new}(\mathtt{a}, \mathtt{Y}_0) \leftarrow ((0.7 \&_{\mathtt{prod}} \mathtt{s}(\mathtt{Y}_0)) \&_{\mathtt{G}} \mathtt{r}(\mathtt{Y}_0))$ with $\alpha = 1$
  $\mathcal{R}_9 : \mathtt{new}(\mathtt{Y}_1, \mathtt{a}) \leftarrow ((0.8 \&_{\mathtt{luka}} \mathtt{r}(\mathtt{Y}_1)) \&_{\mathtt{G}} \mathtt{r}(\mathtt{a}))$ with $\alpha = 1$
- $\mathcal{P}_{\mathbf{4}} = (\mathcal{P}_{\mathbf{3}} - \{\mathcal{R}_{\mathbf{8}}\}) \cup \{\mathcal{R}_{\mathbf{10}}\}$
  $\mathcal{R}_{10} : \mathtt{new}(\mathtt{a}, \mathtt{b}) \leftarrow ((0.7 \&_{\mathtt{prod}} 0.9) \&_{\mathtt{G}} \mathtt{r}(\mathtt{Y}_0))$ with $\alpha = 1$
- $\mathcal{P}_{\mathbf{5}} = (\mathcal{P}_{\mathbf{4}} - \{\mathcal{R}_{\mathbf{10}}\}) \cup \{\mathcal{R}_{\mathbf{11}}\}$
  $\mathcal{R}_{11} : \mathtt{new}(\mathtt{a}, \mathtt{b}) \leftarrow ((0.7 \&_{\mathtt{prod}} 0.9) \&_{\mathtt{G}} 0.6)$ with $\alpha = 1$
- $\mathcal{P}_{\mathbf{6}} = (\mathcal{P}_{\mathbf{5}} - \{\mathcal{R}_{\mathbf{11}}\}) \cup \{\mathcal{R}_{\mathbf{12}}\}$
  $\mathcal{R}_{12} : \mathtt{new}(\mathtt{a}, \mathtt{b}) \leftarrow (0.63 \&_{\mathtt{G}} 0.6)$ with $\alpha = 1$

Fuzzy Transformation Rules

- $\mathcal{P}_{\mathbf{3}} = (\mathcal{P}_{\mathbf{2}} - \{\mathcal{R}_{\mathbf{6}}\}) \cup \{\mathcal{R}_{\mathbf{8}}, \mathcal{R}_{\mathbf{9}}\}$
  $\mathcal{R}_8 : \mathtt{new}(\mathsf{a}, \mathtt{Y}_0) \leftarrow ((0.7 \&_{\mathtt{prod}} \mathtt{s}(\mathtt{Y}_0)) \&_{\mathtt{G}} \mathtt{r}(\mathtt{Y}_0))$ with $\alpha = 1$
  $\mathcal{R}_9 : \mathtt{new}(\mathtt{Y}_1, \mathsf{a}) \leftarrow ((0.8 \&_{\mathtt{luka}} \mathtt{r}(\mathtt{Y}_1)) \&_{\mathtt{G}} \mathtt{r}(\mathsf{a}))$ with $\alpha = 1$
- $\mathcal{P}_{\mathbf{4}} = (\mathcal{P}_{\mathbf{3}} - \{\mathcal{R}_{\mathbf{8}}\}) \cup \{\mathcal{R}_{\mathbf{10}}\}$
  $\mathcal{R}_{10} : \mathtt{new}(\mathsf{a}, \mathsf{b}) \leftarrow ((0.7 \&_{\mathtt{prod}} 0.9) \&_{\mathtt{G}} \mathtt{r}(\mathtt{Y}_0))$ with $\alpha = 1$
- $\mathcal{P}_{\mathbf{5}} = (\mathcal{P}_{\mathbf{4}} - \{\mathcal{R}_{\mathbf{10}}\}) \cup \{\mathcal{R}_{\mathbf{11}}\}$
  $\mathcal{R}_{11} : \mathtt{new}(\mathsf{a}, \mathsf{b}) \leftarrow ((0.7 \&_{\mathtt{prod}} 0.9) \&_{\mathtt{G}} 0.6)$ with $\alpha = 1$
- $\mathcal{P}_{\mathbf{6}} = (\mathcal{P}_{\mathbf{5}} - \{\mathcal{R}_{\mathbf{11}}\}) \cup \{\mathcal{R}_{\mathbf{12}}\}$
  $\mathcal{R}_{12} : \mathtt{new}(\mathsf{a}, \mathsf{b}) \leftarrow (0.63 \&_{\mathtt{G}} 0.6)$ with $\alpha = 1$
- $\mathcal{P}_{\mathbf{7}} = (\mathcal{P}_{\mathbf{6}} - \{\mathcal{R}_{\mathbf{12}}\}) \cup \{\mathcal{R}_{\mathbf{13}}\}$
  $\mathcal{R}_{13} : \mathtt{new}(\mathsf{a}, \mathsf{b}) \leftarrow 0.6$ with $\alpha = 1$

Fuzzy Transformation Rules

- **FACTING RULE**

Let $\mathcal{R} : (A \leftarrow_i r \text{ with } \alpha = v) \in \mathcal{P}_k$ where $r \in L$

$$\mathcal{P}_{\mathbf{k+1}} = (\mathcal{P}_{\mathbf{k}} - \{\mathcal{R}\}) \cup \boxed{\{\mathbf{A} \leftarrow \text{ with } \alpha = [\![\&_{\mathbf{i}}]\!](\mathbf{v}, \mathbf{r})\}}$$

Fuzzy Transformation Rules

- **FACTING RULE**

  Let $\mathcal{R} : (A \leftarrow_i r \text{ with } \alpha = v) \in \mathcal{P}_k$ where $r \in L$

  $$\mathcal{P}_{\mathbf{k+1}} \; = \; (\mathcal{P}_{\mathbf{k}} - \{\mathcal{R}\}) \cup \boxed{\{\mathbf{A} \leftarrow \; \text{with } \alpha = [\![\&_{\mathbf{i}}]\!](\mathbf{v}, \mathbf{r})\}}$$

- **EXAMPLE:** Facting $\mathcal{R}_{13} : \mathtt{new}(\mathtt{a}, \mathtt{b}) \leftarrow 0.6 \text{ with } \alpha = 1$

  Remember that $[\![\&]\!](1, v) = [\![\&]\!](v, 1) = v$ which implies that $[\![\&]\!](1, 0.6) = 0.6$, and hence....

$$\mathcal{P}_{\mathbf{8}} = (\mathcal{P}_{\mathbf{7}} - \{\mathcal{R}_{\mathbf{13}}\}) \cup \{\mathcal{R}_{\mathbf{14}} : \mathbf{new}(\mathbf{a}, \mathbf{b}) \leftarrow \; \text{with } \alpha = \mathbf{0.6} \; \}$$

Fuzzy Transformation Rules

- Final program $\mathcal{P}_{\mathbf{8}} = \{\mathcal{R}_{\mathbf{2}}, \mathcal{R}_{\mathbf{3}}, \mathcal{R}_{\mathbf{4}}, \mathcal{R}_{\mathbf{5}}, \mathcal{R}_{\mathbf{7}}, \mathcal{R}_{\mathbf{9}}, \mathcal{R}_{\mathbf{14}}\}$

Fuzzy Transformation Rules

- Final program $\mathcal{P}_{\mathbf{8}} = \{\mathcal{R}_{\mathbf{2}}, \mathcal{R}_{\mathbf{3}}, \mathcal{R}_{\mathbf{4}}, \mathcal{R}_{\mathbf{5}}, \mathcal{R}_{\mathbf{7}}, \mathcal{R}_{\mathbf{9}}, \mathcal{R}_{\mathbf{14}}\}$

$$\begin{array}{ll}
\langle \underline{p(X)} \&_{\texttt{G}} r(a); id\rangle & \rightarrow_{AS1}{}^{\mathcal{R}_7} \\
\langle (0.8 \&_{\texttt{prod}} \underline{new(X_1, Y_1)}) \&_{\texttt{G}} r(a); \{X/X_1\}\rangle & \rightarrow_{AS2}{}^{\mathcal{R}_{14}} \\
\langle (0.8 \&_{\texttt{prod}} 0.6) \&_{\texttt{G}} \underline{r(a)}; \{X/a, X_1/a, Y_1/b\}\rangle & \rightarrow_{AS2}{}^{\mathcal{R}_4} \\
\langle (\underline{0.8 \&_{\texttt{prod}} 0.6}) \&_{\texttt{G}} 0.6; \{X/a, X_1/a, Y_1/b, Y_2/a\}\rangle & \rightarrow_{IS} \\
\langle \underline{0.48 \&_{\texttt{G}} 0.6}; \{X/a, X_1/a, Y_1/b, Y_2/a\}\rangle & \rightarrow_{IS} \\
\langle 0.48; \{X/a, X_1/a, Y_1/b, Y_2/a\}\rangle &
\end{array}$$

Fuzzy Transformation Rules

- Final program $\mathcal{P}_{\mathbf{8}} = \{\mathcal{R}_{\mathbf{2}}, \mathcal{R}_{\mathbf{3}}, \mathcal{R}_{\mathbf{4}}, \mathcal{R}_{\mathbf{5}}, \mathcal{R}_{\mathbf{7}}, \mathcal{R}_{\mathbf{9}}, \mathcal{R}_{\mathbf{14}}\}$

$$
\begin{array}{ll}
\langle \underline{p(X)} \&_{\texttt{G}} r(a); id\rangle & \rightarrow_{AS1}{}^{\mathcal{R}_7} \\
\langle (0.8 \&_{\texttt{prod}} \underline{new(X_1, Y_1)}) \&_{\texttt{G}} r(a); \{X/X_1\}\rangle & \rightarrow_{AS2}{}^{\mathcal{R}_{14}} \\
\langle (0.8 \&_{\texttt{prod}} 0.6) \&_{\texttt{G}} \underline{r(a)}; \{X/a, X_1/a, Y_1/b\}\rangle & \rightarrow_{AS2}{}^{\mathcal{R}_4} \\
\langle (\underline{0.8 \&_{\texttt{prod}} 0.6}) \&_{\texttt{G}} 0.6; \{X/a, X_1/a, Y_1/b, Y_2/a\}\rangle & \rightarrow_{IS} \\
\langle \underline{0.48 \&_{\texttt{G}} 0.6}; \{X/a, X_1/a, Y_1/b, Y_2/a\}\rangle & \rightarrow_{IS} \\
\langle 0.48; \{X/a, X_1/a, Y_1/b, Y_2/a\}\rangle &
\end{array}
$$

IMPROVEMENT: less derivation steps!!!!

In $\mathcal{P}_0$: 9 steps (5+4) $\Rightarrow$ In $\mathcal{P}_8$: 5 steps (3+2) ($\approx$ 50%)

Fuzzy Transformation Rules

THEOREM: Strong Correctness of the Transformation System

Let $(\mathcal{P}_0, \ldots, \mathcal{P}_k)$ be a transformation sequence such that $\mathcal{P}_j$ is obtained from $\mathcal{P}_{j-1}, 0 < j \leq k$, by definition introduction, folding, unfolding or facting. Then,

$$\boxed{\begin{array}{c}\langle Q; id\rangle \rightarrow^{*}_{AS/IS} \langle r; \theta\rangle \text{ in } \mathcal{P}_0 \text{ iff} \\ \langle Q; id\rangle \rightarrow^{*}_{AS/IS} \langle r; \theta'\rangle \text{ in } \mathcal{P}_k\end{array}}$$

where $r \in L$ and $\theta' = \theta[\mathcal{V}ar(Q)]$.

Conclusions

- Fuzzy Logic Programming LANGUAGE:
  The Multi-Adjoint Logic Programming approach
- Fuzzy (CORRECT) Transformation RULES:
  Definition intr., folding, unfolding and facting
- Fuzzy (EFFICIENT) Transformation STRATEGY:
  Generate and "eureka", link/fold it to $\mathcal{R}_0$ and improve its definition by unfolding/facting

Future work

More LANGUAGES: Functional-Fuzzy-Logic,...
More RULES: Non reversible folding, abstraction,...
More STRATEGIES: Composition, tupling,...