Symbolic Unfolding of Multi-adjoint Logic Programs

Ginés Moreno[1] Jaime Penabad[2] José Antonio Riaza[1]

{Gines.Moreno, Jaime.Penabad, JoseAntonio.Riaza}@uclm.es

[1]Dept. of Computing Systems, UCLM, 02071 Albacete (Spain)

[2]Dept. of Mathematics, UCLM, 02071 Albacete (Spain)

9th European Symposium on Computational Intelligence and Mathematics
October 4th - 7th, 2017, Faro (Portugal)

Index

Index

Fuzzy Logic Programming

Fuzzy Logic

$+$

Logic Programming

$\Downarrow$

Fuzzy Logic Programming

Multi-Adjoint Lattices

Multi-adjoint Lattice

A tuple $(L, \leq, \leftarrow_1, \&_1, \ldots, \leftarrow_n, \&_n)$ is a **multi-adjoint lattice** if it verifies the following claims:

i) $(L, \leq)$ is a **complete lattice**, with $\bot$ and $\top$ elements.

ii) $(\&_i, \leftarrow_i)$ is an **adjoint pair** in $(L, \leq)$, i.e.:

1) $\&_i$ is increasing in both arguments.
2) $\leftarrow_i$ is increasing in the first argument and decreasing in the second one.
3) **Adjoint property**: $x \leq (y \leftarrow_i z) \iff (x \&_i z) \leq y$

Examples of Multi-Adjoint Lattices

- **Boolean values** $\{true, false\}$ with the classical adjoint pair.
- **Real numbers in the unit interval** $[0,1]$ with adjoint pairs from the fuzzy logics of **Product**, **Gödel** and **Łukasiewicz**:

$$\&_{\texttt{prod}}(x,y) \triangleq x \cdot y \qquad \leftarrow_{\texttt{prod}}(x,y) \triangleq \begin{cases} 1 & \text{if } y \leq x \\ x/y & \text{if } 0 < x < y \end{cases}$$

$$\&_{\texttt{godel}}(x,y) \triangleq \min(x,y) \qquad \leftarrow_{\texttt{godel}}(x,y) \triangleq \begin{cases} 1 & \text{if } y \leq x \\ x & \text{otherwise} \end{cases}$$

$$\&_{\texttt{luka}}(x,y) \triangleq \max(0, x+y-1) \qquad \leftarrow_{\texttt{luka}}(x,y) \triangleq \min(x-y+1, 1)$$

We can define other connectives like:

- | **disjunctions**: $|_{\texttt{luka}}(x,y) \triangleq min(1, x+y)$
- @ **aggregators**: $@_{\texttt{aver}}(x,y) \triangleq (x+y)/2$

Multi-Adjoint Logic Programming

- We work with a Prolog-like first order language, with variables, function and predicate symbols, **but more connectives**:

$$\begin{array}{lllll} \&_1, & \&_2, & \ldots, & \&_n & (\textit{conjunctions}) \\ |_1, & |_2, & \ldots, & |_n & (\textit{disjunctions}) \\ \leftarrow_1, & \leftarrow_2, & \ldots, & \leftarrow_n & (\textit{implications}) \\ @_1, & @_2, & \ldots, & @_n & (\textit{aggregations}) \end{array}$$

- Instead of naive $\{true, false\}$, we use a **multi–adjoint lattice** to model **truth degrees** $(L, \leq, \leftarrow_1, \&_1, \ldots, \leftarrow_n, \&_n)$.

$$([0,1], \leq, \leftarrow_{\texttt{luka}}, \&_{\texttt{luka}}, \leftarrow_{\texttt{prod}}, \&_{\texttt{prod}}, \leftarrow_{\texttt{godel}}, \&_{\texttt{godel}})$$

Symbolic Multi-Adjoint Logic Programming

- MALP programs contain **weighted rules** $H \leftarrow_i B$ *with* v:

$$P = \left\{ \begin{array}{lllll} R_1 : & p(X) & \leftarrow_{\texttt{prod}} & q(X) \,\&_{\texttt{godel}}\, @_{\texttt{aver}}(r(X), s(X)) & \textit{with } 0.9. \\ R_2 : & q(a) & \leftarrow & & \textit{with } 0.8. \\ R_3 : & r(X) & \leftarrow & & \textit{with } 0.7. \\ R_4 : & s(X) & \leftarrow & & \textit{with } 0.5. \end{array} \right.$$

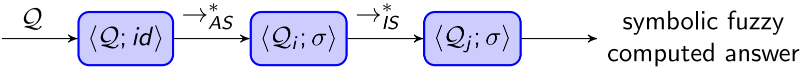
- sMALP programs also allow **symbolic constants**:

$$P^s = \left\{ \begin{array}{lllll} R_1 : & p(X) & \#\!\!\leftarrow^{\texttt{s1}} & q(X) \,\#\&^{\texttt{s2}}\, @_{\texttt{aver}}(r(X), s(X)) & \textit{with } 0.9. \\ R_2 : & q(a) & \leftarrow & & \textit{with } \#^{\texttt{s3}}. \\ R_3 : & r(X) & \leftarrow & & \textit{with } 0.7. \\ R_4 : & s(X) & \leftarrow & & \textit{with } 0.5. \end{array} \right.$$

Index

Operational Semantics for sMALP

- A state is a pair $\langle \mathcal{Q}; \sigma \rangle$, where $\mathcal{Q}$ is a goal and σ a substitution.
- Procedural semantics:
  1. **Operational phase** based on admissible steps $(\to_{AS})$.
  2. **Interpretive phase** based on interpretive steps $(\to_{IS})$.
- Given a program P, a goal $\mathcal{Q}$ and a substitution σ, we define a **state transition system** whose transition relations are $\to_{AS}$ and $\to_{IS}$.



Operational phase (I)

Admissible step using facts

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

1. A is the selected atom in $\mathcal{Q}$,
2. $\theta = mgu(H, A) \neq fail$,
3. $H \leftarrow$ *with* $v \ll P$.

Example

Let $p(a) \leftarrow$ *with* 0.7 be a rule:

$\langle (p(b)\ \&_{\texttt{godel}}\ \underline{p(X)})\ \&_{\texttt{godel}}\ q(X); id \rangle \rightarrow_{AS}$

$\langle (p(b)\ \&_{\texttt{godel}}\ \underline{0.7})\ \&_{\texttt{godel}}\ q(a); \{X/a\} \rangle$

Operational phase (II)

Admissible step using non-empty rules

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

1. A is the selected atom in $\mathcal{Q}$,
2. $\theta = mgu(H, A) \neq fail$,
3. $H \leftarrow_i \mathcal{B}$ *with* $v \ll P$ and $\mathcal{B}$ is not empty.

Example

Let $p(a) \leftarrow_{\texttt{prod}} p(f(a))$ *with* 0.7 be a rule:

$\langle (p(b) \;\&_{\texttt{godel}}\; \underline{p(X)}) \;\&_{\texttt{godel}}\; q(X); id\rangle \rightarrow_{AS}$

$\langle (p(b) \;\&_{\texttt{godel}}\; \underline{0.7 \;\&_{\texttt{prod}}\; p(f(a))}) \;\&_{\texttt{godel}}\; q(a); \{X/a\}\rangle$

Operational phase (III)

Admissible step not using program rules

$\langle \mathcal{Q}[A]; \sigma \rangle \rightarrow_{AS} \langle \mathcal{Q}[A/\bot]; \sigma \rangle$ if

1. A is the selected atom in $\mathcal{Q}$,
2. there is no rule in P whose head unifies with A.

Example

This case is introduced to cope with (possible) unsuccessful admissible derivations. So, with program $p(a) \leftarrow with\ 0.7$ we have:

$\langle \underline{p(b)}; id \rangle \rightarrow_{AS} \langle \underline{0}; id \rangle$

Interpretive phase

Interpretive step

$\langle \mathcal{Q}[\zeta(r_1, \ldots, r_n)]; \sigma \rangle \rightarrow_{IS} \langle \mathcal{Q}[\zeta(r_1, \ldots, r_n)/r_{n+1}]; \sigma \rangle$

where ζ denotes a connective defined on the lattice L associated to P and $\zeta(r_1, \ldots, r_n) \triangleq r_{n+1}$.

Example

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

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

$\langle (0.8 \ \&_{\texttt{luka}} \ (\underline{0.63} \ \&_{\texttt{godel}} \ 0.7)); \{X/a\} \rangle$

Example of derivation

$$P = \begin{cases} R_1 : & p(X) & \#\leftarrow^{s1} & q(X)\ \#\&^{s2}\ @_{\mathtt{aver}}(r(X), s(X)) & \textit{with } 0.9. \\ R_2 : & q(a) & \leftarrow & & \textit{with } \#^{s3}. \\ R_3 : & r(X) & \leftarrow & & \textit{with } 0.7. \\ R_4 : & s(X) & \leftarrow & & \textit{with } 0.5. \end{cases}$$

$$\begin{array}{ll} \langle \underline{p(X)}; id \rangle & \rightarrow^{R_1}_{AS} \\ \langle 0.9\ \#\&^{s1}\ (\underline{q(X_1)}\ \#\&^{s2}\ @_{\mathtt{aver}}(r(X_1), s(X_1))); \{X/X_1\} \rangle & \rightarrow^{R_2}_{AS} \\ \langle 0.9\ \#\&^{s1}\ (\#^{s3}\ \#\&^{s2}\ @_{\mathtt{aver}}(\underline{r(a)}, s(a))); \{X/a\} \rangle & \rightarrow^{R_3}_{AS} \\ \langle 0.9\ \#\&^{s1}\ (\#^{s3}\ \#\&^{s2}\ @_{\mathtt{aver}}(0.7, \underline{s(a)})); \{X/a\} \rangle & \rightarrow^{R_4}_{AS} \\ \langle 0.9\ \#\&^{s1}\ (\#^{s3}\ \#\&^{s2}\ \underline{@_{\mathtt{aver}}(0.7, 0.5)}); \{X/a\} \rangle & \rightarrow_{IS} \\ \langle 0.9\ \#\&^{s1}\ (\#^{s3}\ \#\&^{s2}\ 0.6); \{X/a\} \rangle & \end{array}$$

Index

Symbolic Unfolding

- It is usually based on the **application of computational steps** on the body of program rules.
- It is able to improve programs, generating **more efficient code while preserving the semantics** of the initial program.

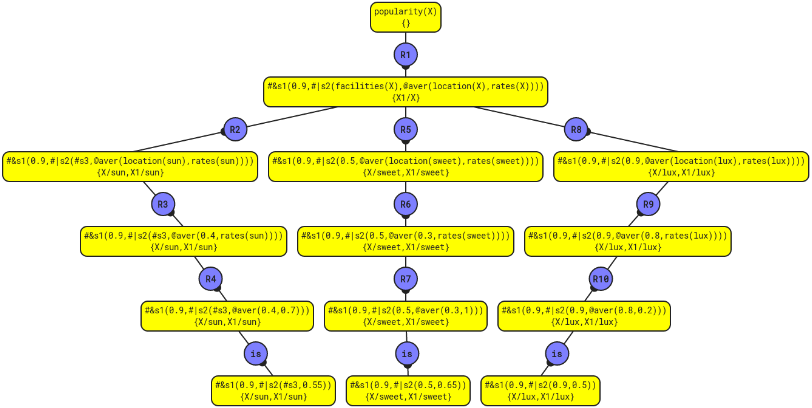
Symbolic Unfolding

Let P be an sMALP program and let $R : A \leftarrow \mathcal{B} \in P$ be a program rule with no empty body. Then, the symbolic unfolding of rule R in program P is the new sMALP program $P' = (P - \{R\}) \cup \{A\sigma \leftarrow \mathcal{B}' \mid \langle \mathcal{B}; id \rangle \to \langle \mathcal{B}'; \sigma \rangle\}$.

Example of Symbolic Unfolding (I)

$$P_0 = \left\{ \begin{array}{lrll}
R_1 : & popularity(X) & \#{\leftarrow}^{s1}\ facilities(X)\ \#|^{s2} & \\
 & & @_{aver}(location(X), rates(X)) & with\ 0.9. \\
 & & & \\
R_2 : & facilities(sun) & \leftarrow & with\ \#^{s3}. \\
R_3 : & location(sun) & \leftarrow & with\ 0.4. \\
R_4 : & rates(sun) & \leftarrow & with\ 0.7. \\
 & & & \\
R_5 : & facilities(sweet) & \leftarrow & with\ 0.5. \\
R_6 : & location(sweet) & \leftarrow & with\ 0.3. \\
R_7 : & rates(sweet) & \leftarrow & with\ 1.0. \\
 & & & \\
R_8 : & facilities(lux) & \leftarrow & with\ 0.9. \\
R_9 : & location(lux) & \leftarrow & with\ 0.8. \\
R_{10} : & rates(lux) & \leftarrow & with\ 0.2.
\end{array} \right.$$

Example of Symbolic Unfolding (II)



Example of Symbolic Unfolding (III)

Unfolding rule R_1 with selected atom $facilities(X)$ applying an $\rightarrow_{AS}$ step with the facts R_2, R_5 and R_8.

1 $popularity(X)\ \#\!\leftarrow^{s1}\ \underline{facilities(X)}\ \#|^{s2}\ @_{\mathtt{aver}}(location(X), rates(X))$ *with* 0.9.

1-2 $popularity(sun)\ \#\!\leftarrow^{s1}\ \underline{\#^{s3}}\ \#|^{s2}\ @_{\mathtt{aver}}(location(sun), rates(sun))$ *with* 0.9.

1-5 $popularity(sweet)\ \#\!\leftarrow^{s1}\ \underline{0.5}\ \#|^{s2}\ @_{\mathtt{aver}}(location(sweet), rates(sweet))$ *with* 0.9.

1-8 $popularity(lux)\ \#\!\leftarrow^{s1}\ \underline{0.9}\ \#|^{s2}\ @_{\mathtt{aver}}(location(lux), rates(lux))$ *with* 0.9.

Example of Symbolic Unfolding (IV)

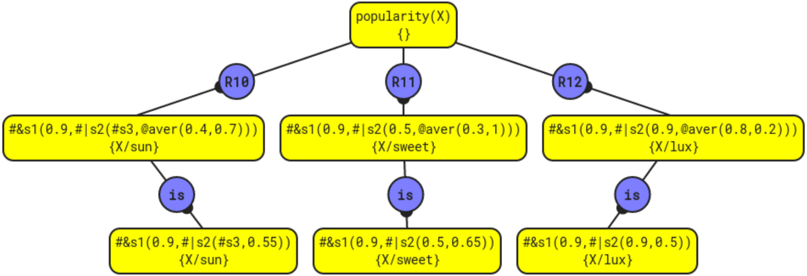
$$P_1 = \left\{ \begin{array}{rrll}
R_{1-2}: & popularity(sun) & \#\!\leftarrow^{s1} \#^{s3}\ \#|^{s2} & \\
 & & @_{aver}(location(sun), rates(sun)) & with\ 0.9. \\
R_{1-5}: & popularity(sweet) & \#\!\leftarrow^{s1} 0.5\ \#|^{s2} & \\
 & & @_{aver}(location(sweet), rates(sweet)) & with\ 0.9. \\
R_{1-8}: & popularity(lux) & \#\!\leftarrow^{s1} 0.9\ \#|^{s2} & \\
 & & @_{aver}(location(lux), rates(lux)) & with\ 0.9. \\
R_2: & facilities(sun) & \leftarrow & with\ \#^{s3}. \\
R_3: & location(sun) & \leftarrow & with\ 0.4. \\
R_4: & rates(sun) & \leftarrow & with\ 0.7. \\
R_5: & facilities(sweet) & \leftarrow & with\ 0.5. \\
R_6: & location(sweet) & \leftarrow & with\ 0.3. \\
R_7: & rates(sweet) & \leftarrow & with\ 1.0. \\
R_8: & facilities(lux) & \leftarrow & with\ 0.9. \\
R_9: & location(lux) & \leftarrow & with\ 0.8. \\
R_{10}: & rates(lux) & \leftarrow & with\ 0.2.
\end{array} \right.$$

Example of Symbolic Unfolding (V)

After aplying 6 unfolds based on $\rightarrow_{AS}$ steps, we have the following program.

$$P_7 = \left\{ \begin{array}{lrlr} R_{1-2-3-4}: & popularity(sun) & \#{\leftarrow}^{s1}\ \#^{s3}\ \#|^{s2}\ @_{aver}(0.4, 0.7) & with\ 0.9. \\ R_{1-5-6-7}: & popularity(sweet) & \#{\leftarrow}^{s1}\ 0.5\ \#|^{s2}\ @_{aver}(0.3, 1.0) & with\ 0.9. \\ R_{1-8-9-10}: & popularity(lux) & \#{\leftarrow}^{s1}\ 0.9\ \#|^{s2}\ @_{aver}(0.8, 0.2) & with\ 0.9. \\ & & & \\ R_2: & facilities(sun) & \leftarrow & with\ \#^{s3}. \\ R_3: & location(sun) & \leftarrow & with\ 0.4. \\ R_4: & rates(sun) & \leftarrow & with\ 0.7. \\ & & & \\ R_5: & facilities(sweet) & \leftarrow & with\ 0.5. \\ R_6: & location(sweet) & \leftarrow & with\ 0.3. \\ R_7: & rates(sweet) & \leftarrow & with\ 1.0. \\ & & & \\ R_8: & facilities(lux) & \leftarrow & with\ 0.9. \\ R_9: & location(lux) & \leftarrow & with\ 0.8. \\ R_{10}: & rates(lux) & \leftarrow & with\ 0.2. \end{array} \right.$$

Example of Symbolic Unfolding (VI)

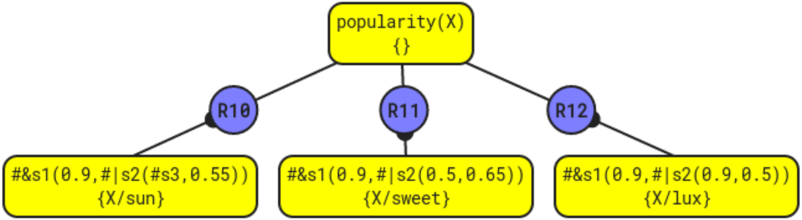


Example of Symbolic Unfolding (VII)

After aplying 3 unfolds based on $\rightarrow_{IS}$ steps, we have the following program.

$$
P_{10} = \left\{ \begin{array}{rrlr}
R_{1-2-3-4-I}: & popularity(sun) & \#\!\leftarrow^{s1} \#^{s3}\ \#|^{s2}\ 0.55 & with\ 0.9. \\
R_{1-5-6-7-I}: & popularity(sweet) & \#\!\leftarrow^{s1} 0.5\ \#|^{s2}\ 0.65 & with\ 0.9. \\
R_{1-8-9-10-I}: & popularity(lux) & \#\!\leftarrow^{s1} 0.9\ \#|^{s2}\ 0.5 & with\ 0.9. \\
\\
R_2: & facilities(sun) & \leftarrow & with\ \#^{s3}. \\
R_3: & location(sun) & \leftarrow & with\ 0.4. \\
R_4: & rates(sun) & \leftarrow & with\ 0.7. \\
\\
R_5: & facilities(sweet) & \leftarrow & with\ 0.5. \\
R_6: & location(sweet) & \leftarrow & with\ 0.3. \\
R_7: & rates(sweet) & \leftarrow & with\ 1.0. \\
\\
R_8: & facilities(lux) & \leftarrow & with\ 0.9. \\
R_9: & location(lux) & \leftarrow & with\ 0.8. \\
R_{10}: & rates(lux) & \leftarrow & with\ 0.2.
\end{array} \right.
$$

Example of Symbolic Unfolding (VIII)



Index

Conclusions and Further Research

Conclusions

- Fuzzy Logic Programming language: sMALP.
- Unfolding sMALP programs improve the efficiency.

Ongoing work

- Proving correctness of the transformation.
- Unfolding symbolic fuzzy programs managing similarity relations.
- Implementing symbolic unfolding in the FLOPER environment.
- Exploring the synergies between symbolic unfolding and tuning transformations.