Fold/Unfold Transformation of Knowledge Bases under Approximate Reasoning

Gabriel Valiente

Institut d'Investigació en Intel·ligència Artificial

Centre d'Estudis Avançats de Blanes

Camí de Santa Bàrbara, 17300 Blanes, Spain

valiente@ceab.es

Abstract

Folding and unfolding are two key program transformation techniques of wide application in logic programming. We present in this paper a natural extension of folding and unfolding for knowledge bases under approximate reasoning that preserves semantics of the knowledge base if the uncertainties of the involved rules satisfy given equations. The equations both state the minimal conditions governing the folding and unfolding of uncertain rules and define the result of the fold resp. unfold operation.

Keywords: Rule-based expert systems, approximate reasoning, knowledge base transformation, folding, unfolding.

1 Introduction

Folding is a program transformation technique that in essence is the inverse of unfolding, a form of partial evaluation of logic programs. Given a logic program P containing the two rules

$$R : B_1, \ldots, B_m, \ldots, B_n \rightarrow A_1$$

and

$$S : B_1, \ldots, B_m \rightarrow A_2$$

the folding of R and S is the new rule

$$R \uparrow S : A_2, B_{m+1}, \ldots, B_n \rightarrow A_1$$

and the replacement of rule R by rule $R \uparrow S$ preserves the semantics of logic program P as long as no rule for A_2 other than S exists in P [4].

Correctness of folding in logic programming rests upon the closed-world assumption. The requirement above that no rule for A_2 other than S exists in P is equivalent to the sentence $B_1, \ldots, B_m \leftrightarrow A_2$ being logical consequence of the completion of P [2].

Unfolding is another program transformation technique of wide application in logic programming. Given a logic program P containing the two rules

$$R : B_1, \ldots, B_m, \ldots, B_n \rightarrow A_1$$

and

$$S : C_1, \ldots, C_k \rightarrow B_m$$

the unfolding of R and S is the resolvent rule

$$R \downarrow S : B_1, \ldots, B_{m-1}, C_1, \ldots, C_k, B_{m+1}, \ldots, B_n \rightarrow A_1$$

and the replacement of rule R by rule $R \downarrow S$ preserves the semantics of logic program P, because it is only doing resolution.

Further assumptions have to be made in order to extend folding and unfolding to other representation languages. We discuss in the following section the minimal assumptions underlying an extension of folding to deal with approximate reasoning, as found in real-world rule-based expert systems.

2 Folding Uncertain Knowledge Bases

A natural extension of folding to knowledge bases under approximate reasoning can be obtained by considering uncertain rules. In order to achieve generality, we make no assumptions regarding the nature of the certainty value L_R associated to each rule R. These values can then stand for certainty values in any particular uncertainty model used.

Given a knowledge base K containing the two rules

$$R : B_1, \ldots, B_m, \ldots, B_n \rightarrow A_1 \ (L_R)$$

and

$$S : B_1, \ldots, B_m \rightarrow A_2 \ (L_S)$$

we define the folding of R and S under approximate reasoning to be the new rule

$$R \uparrow S : A_2, B_{m+1}, \ldots, B_n \rightarrow A_1 \ (L_{R \uparrow S})$$

and want to find the conditions under which the replacement of rule R by rule $R \uparrow S$ preserves the semantics of the knowledge base K, besides the natural requirement that no rule for A_2 other than S exists in K.

It is evident that folding $R \uparrow S$ preserves the semantics of K if, for any assignment of certainty values, the certainty for A_1 obtained through R is identical to that obtained through S and $R \uparrow S$. We proceed then to compute these certainties. In the following, $\triangleright$ stands for the uncertainty propagation (implication) operator and $\triangle$ stands for the conjunction operator.

The certainty for A_1 obtained through R is

$$V(A_1) = \rhd(L_R, \triangle(V(B_1), \ldots, V(B_n))).$$

On the other hand, the certainty for A_1 obtained through S and $R \uparrow S$ is

$$V'(A_1) = \rhd(L_{R\uparrow S}, \triangle(V(A_2), V(B_{m+1}), \ldots, V(B_n)))$$

where the certainty for A_2 is

$$V(A_2) = \rhd(L_S, \triangle(V(B_1), \ldots, V(B_m))).$$

It follows then that

$$V'(A_1) = \rhd(L_{R\uparrow S}, \triangle(\rhd(L_S, \triangle(V(B_1), \ldots, V(B_m))), V(B_{m+1}), \ldots, V(B_n))).$$

In order to be $V(A_1) = V'(A_1)$ it must be the case that

$$\rhd(L_R, \triangle(V(B_1), \ldots, V(B_n))) =$$

$$\rhd(L_{R\uparrow S}, \triangle(\rhd(L_S, \triangle(V(B_1), \ldots, V(B_m))), V(B_{m+1}), \ldots, V(B_n))).$$

The equation can be simplified assuming that $\rhd$ and $\triangle$ are identical, as is true for instance of the MILORD system [1, 3]. We also make use of the associativity of $\triangle$.

$$\begin{aligned}
\triangle(L_R, \triangle(V(B_1), \ldots, V(B_n))) &= \triangle(L_{R\uparrow S}, \triangle(\triangle(L_S, \triangle(V(B_1), \ldots, V(B_m))), \\
&\quad V(B_{m+1}), \ldots, V(B_n))) \\
\triangle(L_R, V(B_1), \ldots, V(B_n)) &= \triangle(L_{R\uparrow S}, L_S, V(B_1), \ldots, V(B_n)) \\
L_R &= \triangle(L_{R\uparrow S}, L_S)
\end{aligned}$$

The previous result provides the foundation for knowledge base folding under approximate reasoning. The requirement for a semantics-preserving folding $R \uparrow S$ expressed by the equation above is that $L_{R \uparrow S}$ be chosen such that $L_R = \Delta(L_{R \uparrow S}, L_S)$. As $L_{R \uparrow S}$ is, in general, not unique, we take the supremum $L_{R \uparrow S} = max\{L | \Delta (L, L_S) = L_R\}$. In those systems where the conjunction operator is taken to be the minimum function, for instance, the equation also states that folding can only be done if $L_S \geq L_R$.

3 Unfolding Uncertain Knowledge Bases

We develop here a natural extension of unfolding to knowledge bases under approximate reasoning, in the same spirit of the extension of folding made in the previous section.

Given a knowledge base K containing the two rules

$$R : B_1, \ldots, B_m, \ldots, B_n \rightarrow A_1 \ (L_R)$$

and

$$S : C_1, \ldots, C_k \rightarrow B_m \ (L_S)$$

we define the unfolding of R and S under approximate reasoning to be the new rule

$$R \downarrow S : B_1, \ldots, B_{m-1}, C_1, \ldots, C_k, B_{m+1}, \ldots, B_n \rightarrow A_1 \ (L_{R \downarrow S})$$

We want to find then the conditions under which the replacement of rule R by rule $R \downarrow S$ preserves the semantics of the knowledge base.

It is evident that unfolding $R \downarrow S$ preserves the semantics of K if, for any interpretation, the certainty for A_1 obtained through R and S is identical to that obtained through $R \downarrow S$. We proceed then to compute these certainties.

The certainty for A_1 obtained through R and S is

$$V(A_1) = \rhd(L_R, \Delta(V(B_1), \ldots, V(B_m), \ldots, V(B_n)))$$

where the certainty for B_m is

$$V(B_m) = \rhd(L_S, \triangle(V(C_1), \ldots, V(C_k)))$$

It follows then that

$$\begin{aligned} V(A_1) &= \rhd(L_R, \triangle(V(B_1), \ldots, V(B_{m-1}), \\ &\quad \rhd(L_S, \triangle(V(C_1), \ldots, V(C_k))), \\ &\quad V(B_{m+1}), \ldots, V(B_n))) \end{aligned}$$

On the other hand, the certainty for A_1 obtained through $R \downarrow S$ is

$$\begin{aligned} V'(A_1) &= \rhd(L_{R\downarrow S}, \triangle(V(B_1), \ldots, V(B_{m-1}), \\ &\quad V(C_1), \ldots, V(C_k), \\ &\quad V(B_{m+1}), \ldots, V(B_n))) \end{aligned}$$

In order to be $V(A_1) = V'(A_1)$ it must be the case that

$$\begin{aligned} &\rhd(L_R, \triangle(V(B_1), \ldots, V(B_{m-1}), \\ &\quad \rhd(L_S, \triangle(V(C_1), \ldots, V(C_k))), \\ &\qquad V(B_{m+1}), \ldots, V(B_n))) = \rhd(L_{R\downarrow S}, \triangle(V(B_1), \ldots, V(B_{m-1}), \\ &\qquad\qquad V(C_1), \ldots, V(C_k), \\ &\qquad\qquad V(B_{m+1}), \ldots, V(B_n))) \end{aligned}$$

The equation can be simplified under the assumption that $\rhd$ and $\triangle$ are identical. We also make use of the associativity, commutativity, and existence of neutral element of $\triangle$.

$$\triangle(L_R, \triangle(V(B_1), \ldots, V(B_{m-1}),$$

$$
\begin{aligned}
\Delta(L_S, \Delta(V(C_1), \ldots, V(C_k)),& \\
V(B_{m+1}), \ldots, V(B_n))) &= \Delta(L_{R\downarrow S}, \Delta(V(B_1), \ldots, V(B_{m-1}), \\
&\quad V(C_1), \ldots, V(C_k), \\
&\quad V(B_{m+1}), \ldots, V(B_n)))
\end{aligned}
$$

$$
\begin{aligned}
\Delta(L_R, V(B_1), \ldots, V(B_{m-1}),& \\
L_S, V(C_1), \ldots, V(C_k),& \\
V(B_{m+1}), \ldots, V(B_n)) &= \Delta(L_{R\downarrow S}, V(B_1), \ldots, V(B_{m-1}), \\
&\quad V(C_1), \ldots, V(C_k), \\
&\quad V(B_{m+1}), \ldots, V(B_n))
\end{aligned}
$$

$$
\begin{aligned}
\Delta(L_R, L_S, V(B_1), \ldots, V(B_{m-1}),& \\
V(C_1), \ldots, V(C_k),& \\
V(B_{m+1}), \ldots, V(B_n)) &= \Delta(L_{R\downarrow S}, V(B_1), \ldots, V(B_{m-1}), \\
&\quad V(C_1), \ldots, V(C_k), \\
&\quad V(B_{m+1}), \ldots, V(B_n))
\end{aligned}
$$

$$
\begin{aligned}
\Delta(\Delta(L_R, L_S), V(B_1), \ldots, V(B_{m-1}),& \\
V(C_1), \ldots, V(C_k),& \\
V(B_{m+1}), \ldots, V(B_n)) &= \Delta(L_{R\downarrow S}, V(B_1), \ldots, V(B_{m-1}), \\
&\quad V(C_1), \ldots, V(C_k), \\
&\quad V(B_{m+1}), \ldots, V(B_n)) \\
\Delta(L_R, L_S) &= L_{R\downarrow S}
\end{aligned}
$$

The previous result provides the foundation for knowledge base unfolding under approximate reasoning. The requirement for a semantics-preserving unfolding $R \downarrow S$ expressed by the equation above is that $L_{R\downarrow S}$ be chosen equal to $\Delta(L_R, L_S)$.

4 Conclusion

We have developed semantics-preserving folding and unfolding operations for knowledge bases under approximate reasoning. The operations are applicable whenever the uncertainties of the involved rules satisfy given equations. The equations also define the result of the folding resp. unfolding operation.

Potential applications of folding and unfolding under approximate reasoning include knowledge base verification, optimization and refinement in rule-based expert systems.

Acknowledgement

The ideas presented in this paper have evolved through many interactions with Lluís Godo, to whom we are very grateful.

References

[1] L. Godo, R. López de Mántaras, C. Sierra, and A. Verdaguer. MILORD: The Architecture and the Management of Linguistically Expressed Uncertainty. *Int. Journal of Intelligent Systems*, 4(4):471–501, 1989.

[2] C. J. Hogger. *Essentials of Logic Programming.* Oxford University Press, New York, 1990.

[3] R. López de Mántaras, J. Agustí, E. Plaza, and C. Sierra. MILORD: A Fuzzy Expert Systems Shell. In A. Kandel, editor, *Fuzzy Expert Systems*, chapter 15. CRC Press, Boca Raton, Florida, 1991.

[4] H. Tamaki and T. Sato. Unfold/Fold Transformation of Logic Programs. In *Proc. 2nd Int. Logic Programming Conference*, pages 127–138, Uppsala, Sweden, July 1984.