

Online appendix for the paper
*Knowledge Compilation of Logic Programs
 Using Approximation Fixpoint Theory*
 published in Theory and Practice of Logic Programming

BART BOGAERTS and GUY VAN DEN BROECK

Department of Computer Science, KU Leuven, Belgium
 (e-mail: bart.bogaerts@cs.kuleuven.be, guy.vandenbroeck@cs.kuleuven.be)

submitted 29 April 2015; revised 3 July 2015; accepted 14 July 2015

Appendix A Figures

This appendix contains some figures associated with the gear wheels example (Example 4.13). The first figure contains a circuit representation of the parametrised well-founded model of logic program \mathcal{P}_w from Example 4.13.

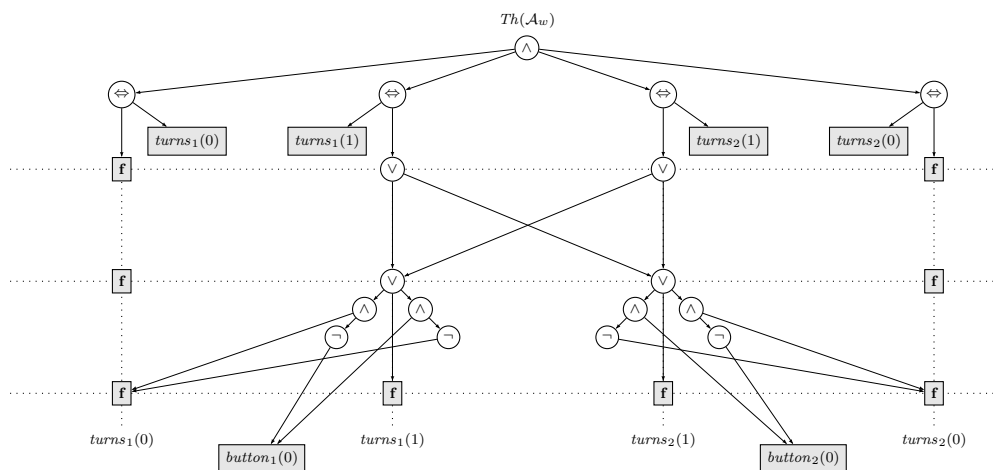


Fig. A 1. A circuit representation of the gear wheel theory $Th(\mathcal{A}_w)$.

The next figure contains a circuit representation of the parametrised well-founded model of the following logic program $\mathcal{P}_{w,2}$ that represent the gear wheel example

with time ranging from 0 to 2:

$$\left\{ \begin{array}{ll} \textit{turns}_1(0) \leftarrow \textit{turns}_2(0) & \textit{turns}_2(0) \leftarrow \textit{turns}_1(0) \\ \textit{turns}_1(1) \leftarrow \textit{turns}_2(1) & \textit{turns}_2(1) \leftarrow \textit{turns}_1(1) \\ \textit{turns}_1(2) \leftarrow \textit{turns}_2(2) & \textit{turns}_2(2) \leftarrow \textit{turns}_1(2) \\ \textit{turns}_1(1) \leftarrow \textit{turns}_1(0) \wedge \neg \textit{button}_1(0) & \textit{turns}_2(1) \leftarrow \textit{turns}_2(0) \wedge \neg \textit{button}_2(0) \\ \textit{turns}_1(1) \leftarrow \neg \textit{turns}_1(0) \wedge \textit{button}_1(0) & \textit{turns}_2(1) \leftarrow \neg \textit{turns}_2(0) \wedge \textit{button}_2(0) \\ \textit{turns}_1(2) \leftarrow \textit{turns}_1(1) \wedge \neg \textit{button}_1(1) & \textit{turns}_2(2) \leftarrow \textit{turns}_2(1) \wedge \neg \textit{button}_2(1) \\ \textit{turns}_1(2) \leftarrow \neg \textit{turns}_1(1) \wedge \textit{button}_1(1) & \textit{turns}_2(2) \leftarrow \neg \textit{turns}_2(1) \wedge \textit{button}_2(1) \end{array} \right\}$$

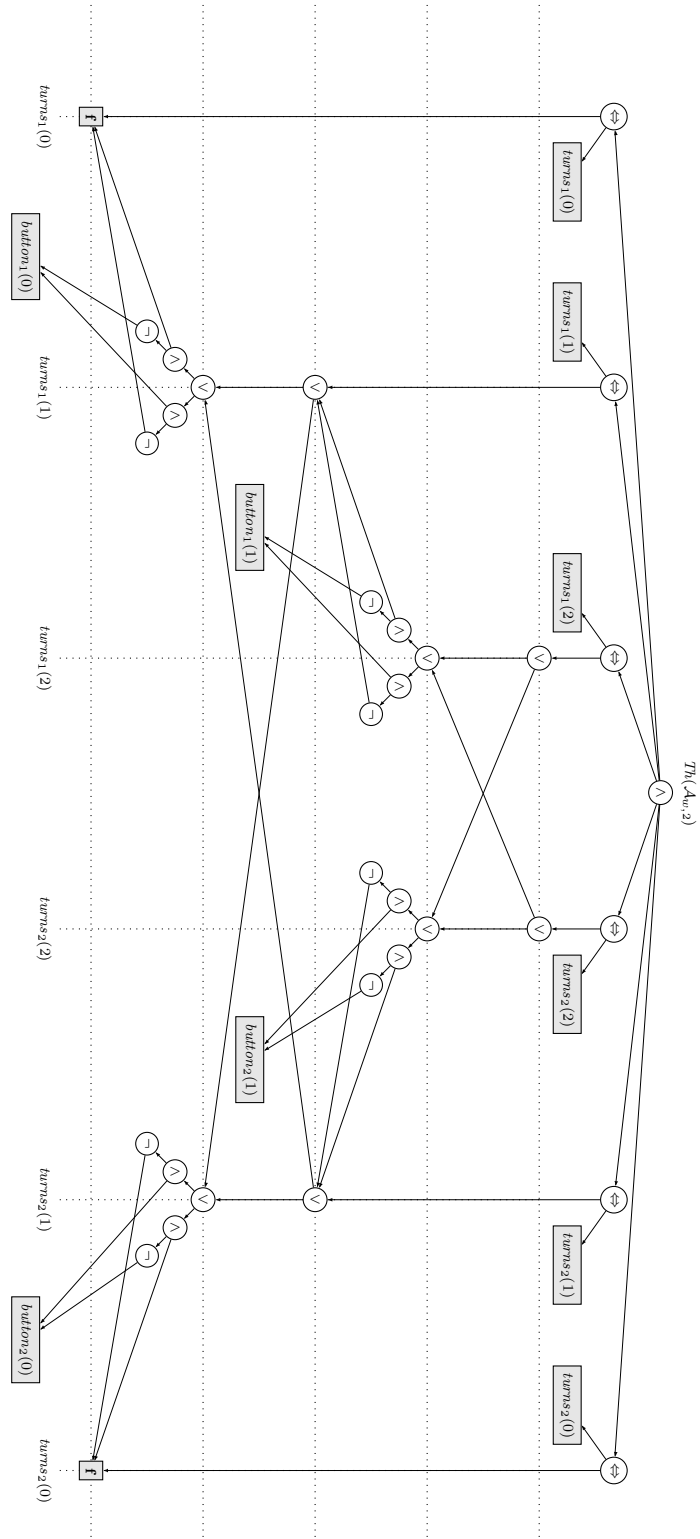


Fig. A 2. A circuit representation of the gear wheel example for up to two time points.

Appendix B Proofs

Definition-Proposition 3.1.

Let $O : L \rightarrow L$ be an operator and $f : L \rightarrow K$ a lattice morphism. We say that O respects f if for every $x, y \in L$ with $f(x) = f(y)$, it holds that $f(O(x)) = f(O(y))$.

If f is surjective and O respects f , then there exists a unique operator $O_f : K \rightarrow K$ with $O_f \circ f = f \circ O$, which we call the *projection of O on K* .

Proof

We prove the existence and uniqueness of O_f .

Choose $x \in K$. Since f is surjective, there is a $x' \in L$ with $f(x') = x$. We know that O_f must map x to $f(O(x'))$, hence uniqueness follows. Furthermore, this mapping is well-defined (independent of the choice of x') since O respects f . \square

Proposition B.1

If (x', y') is an A -refinement of (x, y) , then $(f(x'), f(y'))$ is an A_f -refinement of $(f(x), f(y))$.

Proof

1. First suppose (x', y') is an application A -refinement of (x, y) . Thus

$$(x, y) \leq_p (x', y') \leq_p A(x, y).$$

From the fact that f is a lattice morphism, it follows that

$$f^2(x, y) \leq_p f^2(x', y') \leq_p f^2(A(x, y)).$$

From the fact that f respects A , we then find

$$f^2(x, y) \leq_p f^2(x', y') \leq_p A_f(f^2(x, y)),$$

hence $f^2(x', y')$ is an application A_f -refinement of $f^2(x, y)$.

2. The second direction is analogous to the first. Suppose (x', y') is an unfoundedness A -refinement of (x, y) . Thus $x' = x$ and

$$A(x, y')_2 \leq y' \leq y.$$

Then also $f(x') = f(x)$ and

$$f(A(x, y')_2) \leq f(y') \leq f(y),$$

thus

$$A_f(f(x), f(y'))_2 \leq f(y') \leq f(y)$$

and the result follows.

Lemma B.2

If O and O_f are monotone, then $f(\text{lfp}(O)) = \text{lfp}(O_f)$.

Proof

The least fixpoint of O is the limit of the sequence $\perp \rightarrow O(\perp) \rightarrow O(O(\perp)) \rightarrow \dots$. It follows immediately from the definition of O_f that for every ordinal n , $f(O^n(\perp)) = O_f^n(f(\perp)) = O_f^n(\perp_K)$, hence the result follows. \square

Proposition 3.3.

If $(x_j, y_j)_{j \leq \alpha}$ is a well-founded induction of A , then $(f(x_j), f(y_j))_{j \leq \alpha}$ is a well-founded induction of A_f . If $(x_j, y_j)_{j \leq \alpha}$ is terminal, then so is $(f(x_j), f(y_j))_{j \leq \alpha}$.

Proof

The first claim follows directly (by induction) from Proposition B.1.

For the second claim, all that is left to show is that if there are no strict A -refinements of (x_α, y_α) , then there are also no strict A_f -refinements of $(f(x_\alpha), f(y_\alpha))$.

First of all, since (x_α, y_α) is a fixpoint of A , it also follows for every i that $A_f(f(x_\alpha), f(y_\alpha)) = f^2(A(x_\alpha, y_\alpha)) = (f(x_\alpha), f(y_\alpha))$. Thus, there are no strict application refinements of A_f either.

Since there are no unfoundedness refinements of (x_α, y_α) , Proposition 2.1 yields that $y_\alpha = \text{lfp } S_A^x$. It is easy to see that for every i , the operator $f \circ S_A^x = S_{A_f}^{f(x)} \circ f$. Hence, Lemma B.2 (for the operator S_A^x) guarantees that $f(y_\alpha) = f(\text{lfp } S_A^x) = \text{lfp } S_{A_f}^{f(x)}$. Thus, using Proposition 2.1 we find that there is no strict unfoundedness refinement of $(f(x_\alpha), f(y_\alpha))$.

\square

Theorem 3.4.

If (x, y) is the A -well-founded fixpoint of O , then, $(f(x), f(y))$ is the A_f -well-founded fixpoint of O_f .

Proof

Follows immediately from Proposition 3.3. \square

Theorem 3.6.

Suppose L is a parametrisation of K through $(f_i)_{i \in I}$. Let $O : L \rightarrow L$ be an operator and A an approximator of O such that both O and A respect each of the f_i . If (x, y) is the A -well-founded fixpoint of O , the following hold.

1. For each i , $(f_i(x), f_i(y))$ is the A_{f_i} -well-founded fixpoint of O_{f_i} .
2. If the A_{f_i} -well-founded fixpoint of O_{f_i} is exact for every i , then so is the A -well-founded fixpoint of O .

Proof

The first point immediately follows from Theorem 3.4.

Using the first point, we find that if the A_{f_i} -well-founded fixpoint of O_{f_i} is exact for every i , then $f_i(x) = f_i(y)$ for every i . Hence the definition of parametrisation guarantees that $x = y$ as well, i.e., the A -well-founded fixpoint of O is indeed exact.

□

Proposition 4.5.

For every formula φ over Σ , $\mathcal{S} \in (L_p^d)^2$ and $I \in 2^{\Sigma_p}$, it holds that $\varphi^{\mathcal{S}^I} = (\varphi^{\mathcal{S}})^I$.

Proof

Trivial. □

Proposition 4.6.

The lattice L_p^d is a parametrisation of 2^{Σ_d} through the mappings $(\pi_I : L_p^d \rightarrow 2^{\Sigma_d} : \mathcal{A} \mapsto \mathcal{A}^I)_{I \in 2^{\Sigma_p}}$.

Proof

It is clear that the mappings π_I are lattice morphisms since evaluation of propositional formulas commutes with Boolean operations. Now, for $\mathcal{A}, \mathcal{A}' \in L_p^d$, it holds that $\mathcal{A} \leq \mathcal{A}'$ if and only if for every atom $p \in \Sigma_d$, $\mathcal{A}(p)$ entails $\mathcal{A}'(p)$. This is equivalent to the condition that for every $p \in \Sigma_d$ and every interpretation $I \in 2^{\Sigma_d}$, $\mathcal{A}(p)^I \leq \mathcal{A}'(p)^I$, i.e., with the fact that for every I , $\pi_I(\mathcal{A}) \leq \pi_I(\mathcal{A}')$ which is what we needed to show. □

Theorem 4.8.

If \mathcal{P} is a positive logic program, then $\mathcal{T}_{\mathcal{P}}$ is monotone. For every Σ -interpretation I , it then holds that $I \models_{wf} \mathcal{P}$ if and only if $I \models Th(\text{lfp}(\mathcal{T}_{\mathcal{P}}))$.

Proof

Follows immediately from the definition of the parametrised well-founded semantics combined with Lemma B.2. □

Theorem 4.9.

For any parametrised logic program \mathcal{P} , the following hold:

1. $\Psi_{\mathcal{P}}$ is an approximator of $\mathcal{T}_{\mathcal{P}}$.
2. For every Σ_p -structure I , it holds that $\Psi_{\mathcal{P}}^I \circ \pi_I^2 = \pi_I^2 \circ \Psi_{\mathcal{P}}$.

Proof

1. It follows immediately from the definitions that for exact interpretations $\mathcal{S} = (\mathcal{A}, \mathcal{A})$, $\Psi_{\mathcal{P}}$ coincides with $\mathcal{T}_{\mathcal{P}}$. \leq_p -monotonicity follows directly from the definition of evaluation of formulas (Definition 4.4).
2. We find that for every $\mathcal{S} \in (L_p^d)^2$ and every $p \in 2^{\Sigma_d}$,

$$\begin{aligned} \Psi_{\mathcal{P}}^I(\pi_I^2(\mathcal{S}))(p) &= \Psi_{\mathcal{P}}^I(\mathcal{S}^I)(p) \\ &= \varphi_p^{\mathcal{S}^I} \\ &= (\varphi_p^{\mathcal{S}})^I \\ &= (\Psi_{\mathcal{P}}(\mathcal{S})(p))^I \\ &= \pi_I^2(\Psi_{\mathcal{P}}(\mathcal{S})(p)), \end{aligned}$$

which indeed proves our claim. \square

Lemma B.3

For every Σ_p -interpretation I , there are at most $|\Sigma_d|$ strict refinements in a well-founded induction of $\Psi_{\mathcal{P}}^I$.

Proof

Every strict refinement should at least change one of the atoms in Σ_d from unknown to either true or false, hence the result follows. \square

Lemma B.4

Suppose $(x_i, y_i)_{i \leq \beta}$ is a well-founded induction of $\mathcal{T}_{\mathcal{P}}$ in which every refinement is maximally precise, i.e., either of the form $(x, y) \rightarrow \mathcal{T}_{\mathcal{P}}(x, y)$ or an unfoundedness refinement satisfying the condition in Proposition 2.1. The following hold:

- there are at most $|\Sigma_d|$ subsequent strict application refinements in $(x_i, y_i)_{i \leq \beta}$, and
- if unfoundedness refinements only happen in $(x_i, y_i)_{i \leq \beta}$ when no application refinement is possible, then there are at most $|\Sigma_d|$ unfoundedness refinements.

Proof

For the first part, we notice that every sequence of maximal application refinements maps (by π_I) onto a sequence of maximal application refinements of $\Psi_{\mathcal{P}}^I$. Furthermore, from the proof of Proposition 3.3, it follows that if a $\mathcal{T}_{\mathcal{P}}$ -refinement is strict, then at least one of the induced $\Psi_{\mathcal{P}}^I$ -refinements must be strict as well. The result now follows from Lemma B.3.

The second point is completely similar to the first. There can be at most $|\Sigma_d|$ strict unfoundedness refinements in any well-founded induction of $\Psi_{\mathcal{P}}^I$. Furthermore, the condition in this point guarantees that if for some I , an unfoundedness refinement in the induced well-founded induction is not strict, then neither will any later unfoundedness refinements. Hence, the result follows. \square

Theorem 5.1.

Let \mathcal{L}_{BC} be the language of Boolean circuits. The following hold: (i) $\text{COMPILE}(\mathcal{L}_{BC})$ has polynomial-time complexity and (ii) the size of the output circuit of $\text{COMPILE}(\mathcal{L}_{BC})$ is polynomial in the size of \mathcal{P} .

Proof

First, we notice that if we have a circuit representation of \mathcal{S} , then the representation of $\Psi_{\mathcal{P}}(\mathcal{S})$ consists of the same circuit with maximally three added layers since φ_p is a DNF for every defined atom p (a layer of negations, one of disjunctions and one of conjunctions). Furthermore, the size of these layers is linear in terms of the size of \mathcal{P} . Similarly, the representation of an unfoundedness refinement will only be quadratically in the size of \mathcal{P} (quadratically since computing the smallest y' is a refinement takes a linear number of applications).

The two results now follow from Lemma B.4, which yields a polynomial upper bound on the number of refinements, and which also allows us to ignore the stop conditions (in general checking whether a fixpoint is reached is a co-NP problem, namely checking equivalence of two circuits; however, we do not need to do this since we have an upper bound on the maximal number of refinements before such a fixpoint is reached). \square

Proposition 5.2.

Suppose the parametrised well-founded model of \mathcal{P} is $(\mathcal{A}, \mathcal{A})$. Let $(\mathcal{A}_{i,1}, \mathcal{A}_{i,2})$ be a well-founded induction of $\Psi_{\mathcal{P}}$. Then for every i , $\text{Th}(\mathcal{A}_{i,1}) \models \text{Th}(\mathcal{A}) \models \text{Th}(\mathcal{A}_{i,2})$.

Proof

Denecker and Vennekens (2007) showed that if $(x_i, y_i)_{i \leq \beta}$ is a well-founded induction of A and (x, y) the A -well-founded model of O , then for every $i \leq \beta$, it holds that

$$(x_i, y_i) \leq_p (x, y).$$

Our proposition immediately follows from this result. \square