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 $P_w$ from Example 4.13.

Fig. A1. A circuit representation of the gear wheel theory $Th(A_w)$. 

The next figure contains a circuit representation of the parametrised well-founded model of the following logic program $P_{w,2}$ that represent the gear wheel example.
with time ranging from 0 to 2:

\[
\begin{align*}
\text{turns}_1(0) & \leftarrow \text{turns}_2(0) & \text{turns}_2(0) & \leftarrow \text{turns}_1(0) \\
\text{turns}_1(1) & \leftarrow \text{turns}_2(1) & \text{turns}_2(1) & \leftarrow \text{turns}_1(1) \\
\text{turns}_1(2) & \leftarrow \text{turns}_2(2) & \text{turns}_2(2) & \leftarrow \text{turns}_1(2) \\
\text{turns}_2(1) & \leftarrow \text{turns}_1(0) \land \neg \text{button}_1(0) & \text{turns}_2(1) & \leftarrow \text{turns}_2(0) \land \neg \text{button}_2(0) \\
\text{turns}_1(1) & \leftarrow \neg \text{turns}_1(0) \land \text{button}_1(0) & \text{turns}_2(1) & \leftarrow \neg \text{turns}_2(0) \land \text{button}_2(0) \\
\text{turns}_1(2) & \leftarrow \text{turns}_1(1) \land \neg \text{button}_1(1) & \text{turns}_2(2) & \leftarrow \text{turns}_2(1) \land \neg \text{button}_2(1) \\
\text{turns}_2(2) & \leftarrow \neg \text{turns}_1(1) \land \text{button}_1(1) & \text{turns}_2(2) & \leftarrow \neg \text{turns}_2(1) \land \text{button}_2(1)
\end{align*}
\]
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 $O : L \to L$ be an operator and $f : L \to K$ 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 $O_f : K \to K$ 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 $f$.

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

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

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 $\bot \to O(\bot) \to O(O(\bot)) \to \ldots$. It follows immediately from the definition of $O_f$ that for every ordinal $n$, $f(O^n(\bot)) = O_f(f(\bot)) = O_f(\bot^K)$, hence the result follows. □

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 $(x_\alpha, y_\alpha)$, then there are also no strict $A_f$-refinements of $(f(x_\alpha), f(y_\alpha))$.

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

□

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. □

Theorem 3.6.
Suppose $L$ is a parametrisation of $K$ through $(f_i)_{i \in I}$. Let $O : L \to L$ 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 $O$. 

The first point immediately follows from Theorem 3.4.

Using the first point, we find that if the $A_f$-well-founded fixpoint of $O_f$ 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.

\[\square\]

**Proposition 4.5.**
For every formula $\varphi$ over $\Sigma$, $S \in (L^d_p)^2$ and $I \in 2^{2^\Sigma}$, it holds that $\varphi^S^I = (\varphi^S)^I$.

**Proof**
Trivial. \[\square\]

**Proposition 4.6.**
The lattice $L^d_p$ is a parametrisation of $2^{2^\Sigma}$ through the mappings $(\pi_I : L^d_p \rightarrow 2^{2^\Sigma} : A \mapsto A^I)_{I \in 2^{2^\Sigma}}$.

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

**Theorem 4.8.**
If $P$ is a positive logic program, then $T_P$ is monotone. For every $\Sigma$-interpretation $I$, it then holds that $I \models wf P$ if and only if $I \models Th(lfp(T_P))$.

**Proof**
Follows immediately from the definition of the parametrised well-founded semantics combined with Lemma B.2. \[\square\]

**Theorem 4.9.**
For any parametrised logic program $P$, the following hold:

1. $\Psi_P$ is an approximator of $T_P$.
2. For every $\Sigma_p$-structure $I$, it holds that $\Psi_P^I \circ \pi^I = \pi^I \circ \Psi_P$. 

Proof
1. It follows immediately from the definitions that for exact interpretations $S = (A, A), \Psi_P$ coincides with $T_P \subseteq P$-monotonicity follows directly from the definition of evaluation of formulas (Definition 4.4).
2. We find that for every $S \in (L^d_P)^2$ and every $p \in 2^{\Sigma^d}$,

$$\Psi_P^I(\pi^2_I(S))(p) = \Psi_P^I(S^I)(p) = \varphi^S_I = (\varphi^S_P)^I = (\Psi_P(S)(p))^I = \pi^2_I(\Psi_P(S)(p)),$$

which indeed proves our claim. □

Lemma B.3
For every $\Sigma^d_P$-interpretation $I$, there are at most $|\Sigma^d|$ strict refinements in a well-founded induction of $\Psi_P^I$.

Proof
Every strict refinement should at least change one of the atoms in $\Sigma^d$ from unknown to either true or false, hence the result follows. □

Lemma B.4
Suppose $(x_i, y_i)_{t \leq \beta}$ is a well-founded induction of $T_P$ in which every refinement is maximally precise, i.e., either of the form $(x, y) \rightarrow T_P(x, y)$ 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 $(x_i, y_i)_{t \leq \beta}$, and
- if unfoundedness refinements only happen in $(x_i, y_i)_{t \leq \beta}$ 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 $\pi_I$) onto a sequence of maximal application refinements of $\Psi_P^I$. Furthermore, from the proof of Proposition 3.3, it follows that if a $T_P$-refinement is strict, then at least one of the induced $\Psi_P^I$-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 $\Psi_P^I$. 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. □
Theorem 5.1.
Let $L_{BC}$ be the language of Boolean circuits. The following hold: (i) $\text{Compile}(L_{BC})$ has polynomial-time complexity and (ii) the size of the output circuit of $\text{Compile}(L_{BC})$ is polynomial in the size of $P$.

Proof
First, we notice that if we have a circuit representation of $S$, then the representation of $\Psi_P(S)$ consists of the same circuit with maximally three added layers since $\varphi_p$ 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 $P$. Similarly, the representation of an unfoundedness refinement will only be quadratically in the size of $P$ (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). □

Proposition 5.2.
Suppose the parametrised well-founded model of $P$ is $(A, A)$. Let $(A_{i,1}, A_{i,2})$ be a well-founded induction of $\Psi_P$. Then for every $i$, $Th(A_{i,1}) \models Th(A) \models Th(A_{i,2})$.

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

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

Our proposition immediately follows from this result. □