Well-founded models in proofs of termination

Salvador Lucas

DSIC, Universitat Politècnica de València (UPV)
http://slucas.webs.upv.es/

16th INTERNATIONAL WORKSHOP ON TERMINATION (WST 2018)
Oxford, United Kingdom
Main goal

Understanding, analyzing, and automatically verifying the termination behavior of declarative programs (e.g., Maude)

fmod PALINDROME is
    protecting QID .
    sorts List Pal .
    subsorts Qid < Pal < List .
    op nil : -> Pal .
    op __ : List List -> List .
    var I : Qid .
    var P : Pal .
    mb I P I : Pal .
endfm

fmod PROGRAM is
    sort S .
    ops a b : -> [S] .
    ceq a = b if a : S .
endfm

Challenging!

PALINDROME is (obviously?) terminating. PROGRAM is not (why?).
A theory $S$ in a logic $\mathcal{L}$ is called *operationally terminating* if no infinite *well-formed* proof tree for $S$ exists [LMM05].

*Infinite branches* are captured by using *proof jumps* $A \uparrow B_1, \ldots, B_{n-1}, B_n$ associated to the *inference rules* $\frac{B_1 \cdots B_n \cdots B_p}{A}$ of $S$ [LM14].
Use of logical models and well-founded relations

Provability of goals $\sigma(B_1), \ldots, \sigma(B_{n-1})$ captured by logical models when needed.

The head $A$ and the hook $B_n$ of proof jumps indicate where well-founded relations $\sqsubset$ are required to prove termination: $\sigma(A) \sqsubset \sigma(B_n)$

Termination proofs = Logical models + Well-founded relations
Well-founded models in proofs of termination

Outline

1. Operational termination in General Logics
2. Operational termination and well-founded relations
3. Operational termination and well-founded models
4. Use in the Operational Termination Framework
5. Conclusions and future work
Operational termination in General Logics
A logical approach to declarative programming [Mes87]

1. **Declarative programs** $S$ are *theories* of a given *logic* $\mathcal{L}$.
2. **Computations** with $S$ are implemented as *deductions* in $\mathcal{L}$.
3. **Deductions** proceed according to the *Inference System* $\mathcal{I}(\mathcal{L})$ of $\mathcal{L}$.

**Definition (Proof jumps of a theory $S$ [LM14])**

$$\mathcal{I}_S = \{(A \uparrow \bar{B}_i) \mid \frac{\bar{B}_n}{A} \in \mathcal{I}(S), 1 \leq i \leq n\}$$
The specialization of the inference system for CS-MRTs [DLM+08] yields $I(PALINDROME)$ with 16 rules.

**Subject Reduction** rules concern the dependency of sorts with *rewritings*

$$(SR)_Q \quad x \to y \quad y : \text{Qid} \quad (SR)_P \quad x \to y \quad y : \text{Pal} \quad (SR)_L \quad x \to y \quad y : \text{List}$$

**Membership** rules describe the association of sorts to expressions by means of conditional memberships in the program.

$$(M1)_Q < P \quad x :: \text{Qid} \quad (M1)_P < L \quad x :: \text{Pal} \quad (M1)_Q \quad c :: \text{Qid} \quad \text{for c of sort Qid}$$

$$(M1)_\text{Nil} \quad \text{nil} :: \text{Pal} \quad (M1)_- \quad x :: \text{List} \quad y :: \text{List} \quad (M1)_\text{mbP} \quad l :: \text{Qid} \quad P :: \text{Pal}$$

$$(M2)_Q \quad x :: \text{Qid} \quad (M2)_P \quad x :: \text{Pal} \quad (M2)_L \quad x :: \text{List}$$

**Reflexivity** and **transitivity** rules for the many step rewrite relations.

$$(Rf)_L \quad x \to^* x \quad (T)_L \quad x \to y \quad y \to^* z$$

The **congruence rules** propagate one-step reductions.

$$(C)_- , 1 \quad x \to y \quad x z \to y z \quad (C)_- , 2 \quad x \to y \quad z x \to z y$$
Given $\frac{B_1 \cdots B_n}{A}$ with label $\rho$ and $1 \leq i \leq n$, $[\rho]^i$ is the $i$-th proof jump $A \uparrow B_1, \ldots, B_i$ obtained from $\rho$.

**Example**

The proof jumps obtained from the inference rule

$$\begin{array}{c}
(M1)_{mbP} \\
I :: Qid \\
P :: Pal
\end{array}$$

are the following:

$$
[(M1)_{mbP}]^1 : IPI :: Pal \uparrow I :: Qid
$$

$$
[(M1)_{mbP}]^2 : IPI :: Pal \uparrow I :: Qid, P :: Pal
$$
Operational termination and well-founded relations
Theorem (OT and well-founded relations)

A theory $S$ is operationally terminating iff there is a well-founded relation $\sqsubseteq$ (on formulas) such that, for all $A \uparrow \vec{B}_n \in J_S$,

$$\text{for all } \sigma, \text{ if } S \vdash \sigma(B_i) \text{ for all } i, 1 \leq i < n, \text{ then } \sigma(A) \sqsubseteq \sigma(B_n)$$

(1)

Proof keys:

- **if part** (by contradiction): an infinite well-formed proof tree is represented by an infinite sequence of (instances of) proof jumps which (by hypothesis) lead to an infinite sequence of comparisons with $\sqsubseteq$ and then to a contradiction.

- **only if part**: by operational termination, a well-founded proof progress relation on formulas $\uparrow\uparrow$ is defined by considering all possible well-formed proof trees and the forks we can follow to go up on it. By construction, $\uparrow\uparrow$ can be taken as $\sqsubseteq$ to satisfy (1).
Theorem (OT and well-founded relations)

A theory $S$ is operationally terminating iff there is a well-founded relation $\sqsubseteq$ (on formulas) such that, for all $A \uparrow \vec{B}_n \in \mathcal{J}_S$,

$$\forall \sigma, \text{ if } S \vdash \sigma(B_i) \text{ for all } i, 1 \leq i < n, \text{ then } \sigma(A) \sqsubseteq \sigma(B_n)$$

How to use this result to prove OT? Problems:

1. infinitely many substitutions $\sigma$
2. provability statements $S \vdash \sigma(B_i)$
3. symbol $\sqsubseteq$ does not belong to the language of $S$

Idea

Encode as a first-order satisfiability problem
Operational termination and well-founded models
In the following result, $\mathcal{S}$ is a first-order theory, and

- $\overline{\mathcal{S}}$ is obtained from $\mathcal{I}(\mathcal{S})$ by interpreting inference rules $\frac{B_1 \cdots B_n}{A}$ as first-order sentences $(\forall \overline{x}) B_1 \land \cdots \land B_n \Rightarrow A$.
- $\downarrow$ is a transformation $P(t_1, \ldots, t_n)\downarrow = f_P(t_1, \ldots, t_n)$ from atoms $P(t_1, \ldots, t_n)$ into terms $f_P(t_1, \ldots, t_n)$ of a new sort $s_\tau$, where $f_P : w \rightarrow s_\tau$ are new function symbols for each predicate $P : w$.
- $\pi_\sqsubseteq : s_\tau s_\tau$ is a new binary (infix) predicate accepting terms of sort $s_\tau$.

**Theorem (OT and well-founded models)**

A theory $\mathcal{S}$ is operationally terminating if and only if there is an interpretation $\mathcal{A}$ with no empty domain such that

$$\mathcal{A} \models \overline{\mathcal{S}} \cup \{ B_1 \land \cdots \land B_{n-1} \Rightarrow A\downarrow \pi_\sqsubseteq B_n\downarrow \mid A \uparrow \vec{B}_n \in \mathcal{J}_S \}$$

and $\pi_\mathcal{A}^{\sqsubseteq}$ is well-founded on $\mathcal{A}_{s_\tau}$.
Example (Operational termination of PALINDROME)

We use $S$ instead of $Qid$ to make some symbols ($a$ and $b$) available for sequencing. With AGES [GLR16], a model $\mathcal{A}$ is obtained [LG18]:

$$\mathcal{A}_S = \mathcal{A}_{\text{Pal}} = \mathbb{N} - \{0\} \quad \mathcal{A}_{\text{List}} = \mathbb{N} \quad \mathcal{A}_{s^\tau} = \mathbb{N} \cup \{-1\}$$

Function symbols are interpreted as follows:

\[
\begin{align*}
    a^A &= b^A = 1 \\
    \text{nil}^A &= 1 \\
    f^A_{:S}(x) &= 3x \\
    f^A_{:s}(x) &= -1 \\
    f^A_{\rightarrow}(x, y) &= 2x - 1 \\
    f^A_{\rightarrow^*}(x, y) &= 4x + y + 1
\end{align*}
\]

Finally, predicates are interpreted as follows:

\[
\begin{align*}
    _ : S^A(x) &\iff true \\
    _ : Pal^A(x) &\iff true \\
    _ : List^A(x) &\iff true \\
    _ : s^A(x) &\iff x \geq 1 \\
    _ : \text{Pal}^A(x) &\iff x \geq 1 \\
    _ : \text{List}^A(x) &\iff x \geq 1 \\
    x \rightarrow^A y &\iff x > y \\
    x(\rightarrow^*)^A y &\iff true \\
    x \pi^A y &\iff x > y
\end{align*}
\]
Use in the Operational Termination Framework
A *removal pair* \((\succsim, \sqsubseteq)\), consists of binary relations \(\succsim\) and \(\sqsubseteq\) on formulas such that \(\sqsubseteq\) is well-founded and \(\succsim \circ \sqsubseteq \subseteq \sqsubseteq\) (or \(\sqsubseteq \circ \succsim \subseteq \sqsubseteq\)).

**Definition (Removal pair processor [LM14])**

\(P_{RP}(\mathcal{S}, \mathcal{I}) = \{(\mathcal{S}, \mathcal{I} - \{\psi\})\}\) iff

1. for all \(C \uparrow \bar{D}_m \in \mathcal{I} - \{\psi\}\) and substitutions \(\sigma\), if \(\mathcal{S} \vdash \sigma(D_i)\) for all \(1 \leq i < m\), then \(\sigma(C) \succsim \sigma(D_m)\) or \(\sigma(C) \sqsubseteq \sigma(D_m)\), and

2. for all substitutions \(\sigma\), if \(\mathcal{S} \vdash \sigma(B_i)\) for all \(1 \leq i < n\), then \(\sigma(A) \sqsubseteq \sigma(B_n)\).
Definition (Semantic version of $P_{RP}$ [Luc16])

$$P_{RP}(S, J) = \{(S, J - \{\psi\})\} \text{ if }$$

1. $A \models S$, 
2. if $J - \{\psi\} \neq \emptyset$, then $A \models x \pi \succ y \land y \pi \sqsubseteq z \Rightarrow x \pi \sqsubseteq z$, 
3. for each $C \uparrow \vec{D}_m \in J - \{\psi\}$, there is $\pi \asymp \in \{\pi \succ, \pi \sqsubseteq\}$ such that $A \models \bigwedge_{i=1}^{m-1} D_i \Rightarrow C \downarrow \pi \asymp \vec{D}_m$, and 
4. $\pi \sqsubseteq$ is well-founded and $A \models \bigwedge_{i=1}^{n-1} B_i \Rightarrow A \downarrow \pi \sqsubseteq B_n$
Example (Use of the semantic version of \(P_{RP}\))

Consider the CTRS \(\mathcal{R} = \{a \rightarrow b, f(a) \rightarrow b, g(x) \rightarrow g(a) \iff f(x) \rightarrow x\}\) in [GA01, page 46], where \(\mathcal{I}(\mathcal{R})\) is:

\[
\begin{align*}
(Rf) & \quad x \rightarrow^* x \quad (T) & \quad x \rightarrow y \quad y \rightarrow^* z
\end{align*}
\]

\[
\begin{align*}
(Rl)_1 & \quad a \rightarrow b & (Rl)_2 & \quad f(a) \rightarrow b
\end{align*}
\]

\[
\begin{align*}
(C)_{f,1} & \quad x \rightarrow y \quad f(x) \rightarrow f(y) & (C)_{g,1} & \quad x \rightarrow y \quad g(x) \rightarrow g(y)
\end{align*}
\]

\[
\begin{align*}
(Rl)_3 & \quad f(x) \rightarrow^* x & (RI)_{(\mathcal{T})} & \quad x \rightarrow y \quad y \rightarrow^* z
\end{align*}
\]

We can remove \([(\mathcal{T})]^2\) from the initial OT problem \(\tau_I = (\mathcal{R}, \mathcal{J}_\mathcal{R})\). A model \(\mathcal{A}\) with finite domain \(\mathcal{A} = \{0, 1, 2, 3\}\) is found with Mace4 [McC10]:

\[
\begin{align*}
a^\mathcal{A} &= 0 & b^\mathcal{A} &= 1 & f^\mathcal{A}(x) &= (x + 2) \mod 4 \\
g^\mathcal{A}(x) &= x \mod 2 & f^\mathcal{A}(x, y) &= x \mod 2 & f^\mathcal{A}_\rightarrow(x, y) &= x \mod 2 \\
& \rightarrow^\mathcal{A} &= \{(0, 1), (0, 3), (2, 1), (2, 3)\} & (\rightarrow^*)^\mathcal{A} &= \{(x, x) \mid x \in \mathcal{A}\} \cup \rightarrow^\mathcal{A} \\
& \pi^-\mathcal{A} &= \{(0, 1)\} & (\pi_\geq)^\mathcal{A} &= \{(0, 0), (1, 1)\} \\
& (\pi^+\mathcal{A}) &= \{(0, 1)\}
\end{align*}
\]

Successive applications of \(P_{RP}\) prove operational termination of \(\mathcal{R}\).
Conclusions and future work
Our approach [LM14]:

1. **Declarative programs** $P$ are **theories** of a given **logic** $\mathcal{L}$

2. **Operational termination** characterizes the **termination behavior** of $P$ as the absence of **infinite proof trees** for $P$ (using $I(\mathcal{L})$)

3. **Proof jumps** capture infinite proof trees in computations.

4. The **OT Framework** provides a practical approach to prove or disprove finiteness of OT problems

Termination proofs via logical models and well-founded relations

Logic provides the expressive framework to describe the **language semantics** which is captured by **logical models** when needed

**Proof jumps** indicate where **well-founded** relations are required to prove termination.

Termination proofs $= \text{Logical models} + \text{Well-founded relations}$
In the early literature on termination of (variants of) Term Rewriting Systems, well-founded orderings have been pervasive...

...often as a particular requirement together with those having to do with the specific variant at stake:

<table>
<thead>
<tr>
<th>Variant</th>
<th>Name</th>
<th>Specific requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRS</td>
<td>reduction ordering</td>
<td>monotonicity, stability</td>
</tr>
<tr>
<td>AC-TRS</td>
<td>AC-reduction ordering</td>
<td>compatibility with AC axioms</td>
</tr>
<tr>
<td>CS-TRS</td>
<td>μ-reduction ordering</td>
<td>μ-monotonicity, stability</td>
</tr>
<tr>
<td>CTRS</td>
<td>quasi-decreasingness</td>
<td>subterm property</td>
</tr>
<tr>
<td>CRMS</td>
<td>reductivity</td>
<td>monotonicity, stability, use of ▷</td>
</tr>
</tbody>
</table>

A unified treatment of rewriting-based systems

All these specific requirements are naturally obtained from the logical structure of the programming language semantics.
Challenges

1. More specific techniques for proving operational termination (e.g., dependency pairs) could be formalized/integrated to be used in the OT Framework by using transformations between different logics.

2. The modular structure of programs regarding proofs of operational termination is also worth to be considered in this setting.

3. Definition of proof strategies to improve the efficiency of the proof methodology.

4. Implementation of the OT framework through a combined effort to improve the tools MTT [DLM08] and MU-TERM [AGLN10].
Well-founded models in proofs of termination

Thanks!

F. Durán, S. Lucas, C. Marché, J. Meseguer, X. Urbain,

F. Durán, S. Lucas, and J. Meseguer.

J. Giesl and T. Arts.

S. Lucas.

S. Lucas and R. Gutiérrez.

S. Lucas, C. Marché, and J. Meseguer.

S. Lucas and J. Meseguer.

W. McCune
Prover9 & Mace4.

J. Meseguer.