Degrees of undecidability of in Term Rewriting

Jörg Endrullis, Herman Geuvers, Hans Zantema

Radboud University Nijmegen, Technical University Eindhoven, Free University Amsterdam, The Netherlands

CSL 2009
7-11 September 2009
Coimbra, Portugal

Overview

- Term Rewriting Systems (TRS): definitions and properties
- Overview of results
- The arithmetic and analytical hierarchy
- Technical details
  - Relating Turing Machines to TRSs
  - Classification of properties of Turing Machines
  - Weak Church-Rosser
  - Church-Rosser
  - Dependency Pair Problems

Term Rewriting Systems (TRS): definitions and properties

- A Signature Σ is a finite set of symbols f each having a fixed arity.
- The set $Ter(\Sigma, \mathcal{X})$ of terms is the smallest set satisfying:
  - $\mathcal{X} \subseteq Ter(\Sigma, \mathcal{X})$, and
  - $f(t_1, \ldots, t_n) \in Ter(\Sigma, \mathcal{X})$ if $f \in \Sigma$ with arity n and $\forall i : t_i \in Ter(\Sigma, \mathcal{X})$.
- A term rewriting system (TRS) over Σ, $\mathcal{X}$ is a finite set R of pairs $\langle \ell, r \rangle \in Ter(\Sigma, \mathcal{X})$, called rewrite rules usually written as $\ell \rightarrow r$ for which
  - the left-hand side ℓ is not a variable ($\ell \notin \mathcal{X}$)
  - all variables in the right-hand side r occur in ℓ ($Var(r) \subseteq Var(\ell)$).

Term Rewriting Systems (TRS): definitions and properties

For terms $s, t \in Ter(\Sigma, \mathcal{X})$ we write $s \to_R t$ if there exists a rule $\ell \to r \in R$, a substitution σ and a context ('term with a hole') C such that $s \equiv C[\ell\sigma]$ and $t \equiv C[r\sigma]$

- $\to_R$ is the rewrite relation induced by R,
- $\leftrightarrow_R$ denotes the symmetric, reflexive closure of $\to_R$.
- $\to_R^+$ denotes the transitive closure of $\to_R$.
- $\to_R^*$ denotes the reflexive, transitive closure of $\to_R$.

Basic TRS properties

- R is strongly normalizing (or terminating) on t, denoted $\mathsf{SN}_R(t)$,
  if every rewrite sequence starting from t is finite.
- R is confluent (or Church-Rosser) on t, denoted $\mathsf{CR}_R(t)$,
  if every pair of finite coinitial reductions starting from t can be extended to a common reduct, that is,
  $\forall t_1, t_2.\ t_1 \leftarrow^* t \rightarrow^* t_2 \Rightarrow \exists d.\ t_1 \rightarrow^* d \leftarrow^* t_2$.
- R is weakly confluent (or weakly Church-Rosser) on t, denoted $\mathsf{WCR}_R(t)$, if every pair of coinitial rewrite steps starting from t can be joined, that is,
  $\forall t_1, t_2.\ t_1 \leftarrow t \longrightarrow t_2 \Rightarrow \exists d.\ t_1 \rightarrow^* d \leftarrow^* t_2$.

R is strongly normalizing (SN_R), confluent (CR_R) or weakly confluent (WCR_R) if the respective property holds on all terms $t \in \mathit{Ter}(\Sigma, \mathcal{X})$.

TRS properties

Church-Rosser and Weak Church-Rosser are usually also considered on the ground terms only (ground = closed; no free variables).

- R is ground Church-Rosser, denoted grCR_R, if every pair of finite coinitial reductions starting from any ground t can be extended to a common reduct, that is, $\forall t, t_1, t_2$ ground. $t_1 \leftarrow^* t \rightarrow^* t_2 \Rightarrow \exists d.\, t_1 \rightarrow^* d \leftarrow^* t_2$.
- R is ground weakly Church-Rosser, denoted grWCR_R, if every pair of coinitial rewrite steps starting from a ground t can be joined, that is, $\forall t, t_1, t_2$ ground. $t_1 \leftarrow t \longrightarrow t_2 \Rightarrow \exists d.\, t_1 \rightarrow^* d \leftarrow^* t_2$.

Undecidability of TRS properties

All interesting properties about TRSs are undecidable, but how undecidable?

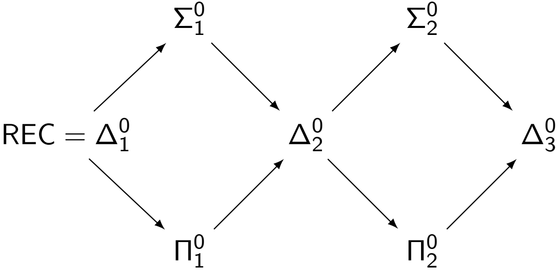
Undecidability of TRS properties

All interesting properties about TRSs are undecidable, but how undecidable?

	SN	WN	CR	grCR	WCR	grWCR	DP	DP^{min}
uniform	Π^0_2	Π^0_2	Π^0_2	Π^0_2	Σ^0_1	Π^0_2	Π^1_1	Π^0_2
single term	Σ^0_1	Σ^0_1	Π^0_2	Π^0_2	Σ^0_1	Σ^0_1	Π^1_1	–

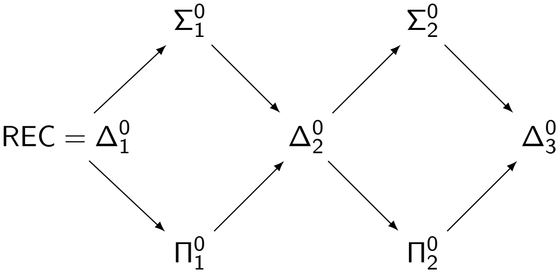
Existing work: Huet and Lankford (1978)
Independent (but published earlier): J.G Simonsen (2009)
New Contributions in red

The Arithmetic Hierarchy



REC = class of decidable problems (over the natural numbers),
$\Sigma_1^0 := \exists\text{REC}$, $\Pi_1^0 := \forall\text{REC}$, $\Sigma_2^0 := \exists\forall\text{REC}$, $\Pi_2^0 := \forall\exists\text{REC}$, etc.

The Arithmetic Hierarchy



REC = class of decidable problems (over the natural numbers),
$\Sigma_1^0 := \exists\text{REC}$, $\Pi_1^0 := \forall\text{REC}$, $\Sigma_2^0 := \exists\forall\text{REC}$, $\Pi_2^0 := \forall\exists\text{REC}$, etc.
$\Delta_n^0 := \Sigma_n^0 \bigcap \Pi_n^0$.
$\Sigma_n^0 = \{A \mid \overline{A} \in \Pi_n^0\}$, $\Pi_n^0 = \{A \mid \overline{A} \in \Sigma_n^0\}$

Examples

We leave encodings implicit, so we say e.g.

- $t \to^* q := \exists \langle s_1, \ldots, s_n \rangle (t \longrightarrow_R s_1 \longrightarrow_R \ldots \longrightarrow_R s_n = q)$ is in Σ^1_0.
- $T(M, \langle \vec{x} \rangle, u, v) :=$ m is a Turing Machine M, u is the computation of M on $\vec{x}$ whose end result is v
  is in REC. Kleene's T-predicate.
- $\mathrm{TOTAL}(M) := \forall x \exists u, v T(m, \langle x \rangle, u, v)$ is in Π^0_2.

Properties of the classes in the Arithmetic Hierarchy

Any formula is equivalent to a formula in prenex normal form

- $Qx\,(\varphi) \otimes Qy\,(\psi) \iff Qx\,Q\,y\,(\varphi \otimes \psi)$, for $\otimes \in \{\wedge, \vee\}$, $Q \in \{\forall, \exists\}$.
- $Qx\,(\varphi) \rightarrow Qy\,(\psi) \iff \overline{Q}x\,Qy\,(\varphi \rightarrow \psi)$, for $Q \in \{\forall, \exists\}$.
  $\iff Qy\,\overline{Q}x\,(\varphi \rightarrow \psi)$.

Properties of the classes in the Arithmetic Hierarchy

Any formula is equivalent to a formula in prenex normal form

- $Qx\,(\varphi) \otimes Qy\,(\psi) \iff Qx\,Q\,y\,(\varphi \otimes \psi)$, for $\otimes \in \{\wedge, \vee\}$, $Q \in \{\forall, \exists\}$.
- $Qx\,(\varphi) \rightarrow Qy\,(\psi) \iff \overline{Q}x\,Qy\,(\varphi \rightarrow \psi)$, for $Q \in \{\forall, \exists\}$.
  $\iff Qy\,\overline{Q}x\,(\varphi \rightarrow \psi)$.

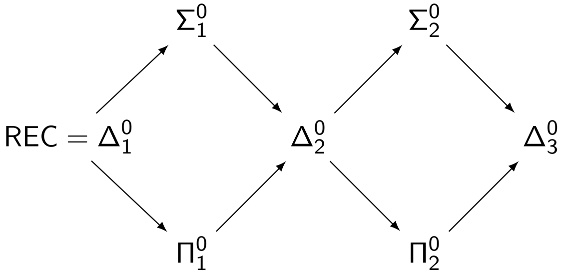
Compression of quantifiers of the same type. Symbolically:

- $\forall\forall \mapsto \forall$ and $\exists\exists \mapsto \exists$
  $\forall x \forall y (P(x, y)) \iff \forall z (P((z)_1, (z)_2))$

A bounded quantifier is no quantifier:

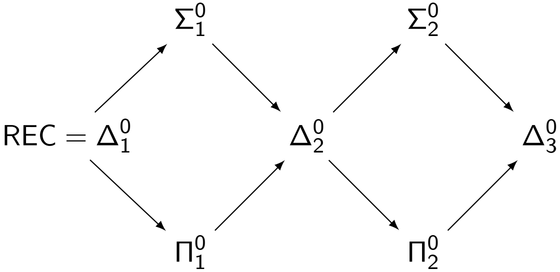
- $\forall x < n\,\mathrm{REC} = \mathrm{REC}$,
- $\exists x < n\,\mathrm{REC} = \mathrm{REC}$

The Arithmetic Hierarchy



Theorem $\Sigma_i^0 \subsetneq \Delta_{i+1}^0 \subsetneq \Sigma_{n+1}^0$ and $\Pi_i^0 \subsetneq \Delta_{i+1}^0 \subsetneq \Pi_{n+1}^0$

The Arithmetic Hierarchy



Theorem $\Sigma_i^0 \subsetneq \Delta_{i+1}^0 \subsetneq \Sigma_{n+1}^0$ and $\Pi_i^0 \subsetneq \Delta_{i+1}^0 \subsetneq \Pi_{n+1}^0$

$\text{BlankTape}(M) := \exists u, v\ T(M, \langle\rangle, u, v) \in \Sigma_1^0 \setminus \Pi_1^0$

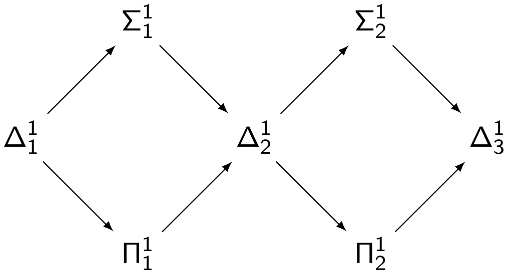
$\text{TOTAL}(M) := \forall x \exists u, v\ T(M, \langle x \rangle, u, v) \in \Pi_2^0 \setminus \Sigma_2^0$

Above the arithmetical hierarchy: analytical hierarchy

All properties definable in first order arithmetic reside in the arithmetical hierarchy.
If we want to quantify over functions from $\mathbb{N}$ to $\mathbb{N}$ (infinite sequences of numbers), we end up in the analytical hierarchy.
Function variables are usually α, β, etc.
Example:

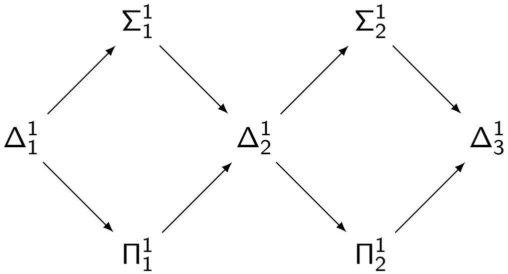
$$\exists\alpha\forall i(\alpha(i) \rightarrow_R \alpha(i+1))$$

The Analytic Hierarchy



$\Sigma^1_1 := \exists\alpha\forall x\text{REC}$, $\Pi^1_1 := \forall\alpha\exists x\text{REC}$, $\Sigma^1_2 := \exists\beta\forall\alpha\exists x\text{REC}$, etc.
$\Delta^1_n := \Sigma^1_n \bigcap \Pi^1_n$.

The Analytic Hierarchy



$\Sigma^1_1 := \exists\alpha\forall x\mathrm{REC}$, $\Pi^1_1 := \forall\alpha\exists x\mathrm{REC}$, $\Sigma^1_2 := \exists\beta\forall\alpha\exists x\mathrm{REC}$, etc.
$\Delta^1_n := \Sigma^1_n \bigcap \Pi^1_n$.
$\Sigma^1_{n+1} = \exists^1\alpha\Pi^1_n$, $\Pi^1_{n+1} = \exists\beta\Sigma^1_n$
$\Sigma^1_n = \{A \mid \overline{A} \in \Pi^1_n\}$, $\Pi^1_n = \{A \mid \overline{A} \in \Sigma^1_n\}$.
Theorem $\Sigma^1_i \subsetneq \Delta^1_{i+1} \subsetneq \Sigma^1_{n+1}$ and $\Pi^1_i \subsetneq \Delta^1_{i+1} \subsetneq \Pi^1_{n+1}$
$\mathrm{WF}(M) :=$ "M defines a well-founded relation $>_M$" $\in \Pi^1_1 \setminus \Sigma^1_1$

Properties of the classes in the Analytic Hierarchy

We have quantifiers over numbers $\forall, \exists$ and over functions $\forall^1, \exists^1$. A number of quantifiers of the same type can be compressed into one.

- $\forall^1\forall^1 \mapsto \forall^1$ and $\exists^1\exists^1 \mapsto \exists^1$

$\forall^1$ subsumes $\forall$.

- $\forall^1\forall \mapsto \forall^1$ and $\exists^1\exists \mapsto \exists^1$

$\forall^1$ moves outside over $\exists$ and $\exists^1$ moves outside over $\forall$.

- $\exists\forall^1 \mapsto \forall^1\exists$ and $\forall\exists^1 \mapsto \exists^1\forall$
- The standard form of an element of the analytic hierarchy is $Q_1^1 Q_2^1 \ldots Q_n^1 Q$ with swopping quantifiers and Q opposite to Q_n^1.

Proving that a property is essentially Π^0_2 (and not "lower")

A total recursive function f many-one reduces problem A to problem B if

$$A(x) \Longleftrightarrow B(f(x)), \text{ for all } x$$

So "if we want to decide $A(x)$, we only have to decide $B(x)$".

$$A \ll_m B \quad (A \text{ is many-one reducible to } B)$$

in case such an f exists.

Proving that a property is essentially Π^0_2 (and not "lower")

A total recursive function f many-one reduces problem A to problem B if

$$A(x) \iff B(f(x)), \text{ for all } x$$

So "if we want to decide $A(x)$, we only have to decide $B(x)$".

$$A \ll_m B \quad (A \text{ is many-one reducible to } B)$$

in case such an f exists.

Definition
B is called Π^0_2-complete if $B \in \Pi^0_2$ and forall $A \in \Pi^0_2$, $A \ll_m B$.
If B is Π^0_2-complete, it can't be lower in the hierarchy.

Proving that a property is essentially Π^0_2 (and not "lower")

A total recursive function f many-one reduces problem A to problem B if

$$A(x) \iff B(f(x)), \text{ for all } x$$

So "if we want to decide $A(x)$, we only have to decide $B(x)$".

$$A \ll_m B \quad (A \text{ is many-one reducible to } B)$$

in case such an f exists.

Definition
B is called Π^0_2-complete if $B \in \Pi^0_2$ and forall $A \in \Pi^0_2$, $A \ll_m B$.
If B is Π^0_2-complete, it can't be lower in the hierarchy.

Theorem
BlankTape(M) is Σ^0_1-complete,
TOTAL(M) is Π^0_2-complete,
WF(M) is Π^1_1-complete.

To prove that WCR is Σ^0_1-complete:
Reduce it to BlankTape

From Turing Machines to TRSs

Translating a Turing machine $M = (Q, \Sigma, q_0, \delta)$ to a TRS R_M
Function symbols:

$a \in \Sigma$	$\mapsto$	unary function $a(-)$
$q \in Q$	$\mapsto$	binary function $q(-,-)$

From Turing Machines to TRSs

Translating a Turing machine $M = (Q, \Sigma, q_0, \delta)$ to a TRS R_M
Function symbols:

$a \in \Sigma$	$\mapsto$	unary function $a(-)$
$q \in Q$	$\mapsto$	binary function $q(-,-)$
extra:		constant $\triangleright$ (representing "infinitely many" blanks)

From Turing Machines to TRSs

Translating a Turing machine $M = (Q, \Sigma, q_0, \delta)$ to a TRS R_M
Function symbols:

$a \in \Sigma$	$\mapsto$	unary function $a(-)$
$q \in Q$	$\mapsto$	binary function $q(-,-)$
extra:		constant $\triangleright$ (representing "infinitely many" blanks)

Configurations:
Right of the reading head: $a\,b\,a\,a\,\Box\,\Box\,\ldots$ translates to
$a(b(a(a(\triangleright))))$
Left of the reading head: $\ldots\,\Box\,\Box\,a\,b\,a\,a$ translates to $a(a(b(a(\triangleright))))$

From Turing Machines to TRSs

Translating a Turing machine $M = (Q, \Sigma, q_0, \delta)$ to a TRS R_M
Function symbols:

$a \in \Sigma$	$\mapsto$	unary function $a(-)$
$q \in Q$	$\mapsto$	binary function $q(-,-)$
extra:		constant $\triangleright$ (representing "infinitely many" blanks)

Configurations:
Right of the reading head: $a\,b\,a\,a\,\Box\,\Box\,\ldots$ translates to
$a(b(a(a(\triangleright))))$
Left of the reading head: $\ldots\,\Box\,\Box\,a\,b\,a\,a$ translates to $a(a(b(a(\triangleright))))$
Tape content $\ldots\,\Box\,w\,\underline{a}\,v\Box\,\ldots$ in state q becomes $q(w^R, a(v))$
(q is reading a, the first symbol of $a\,v$)

Encoding a Turing Machine M as a TRS R_M

Translating the transition function δ:

$$
\begin{array}{rcllcl}
q(x, f(y)) & \longrightarrow & q'(f'(x), y) & \text{if} & \delta(q, f) & = & (q', f', R) \\
q(g(x), f(y)) & \longrightarrow & q'(x, g(f'(y))) & \text{if} & \delta(q, f) & = & (q', f', L)
\end{array}
$$

Encoding a Turing Machine M as a TRS R_M

Translating the transition function δ:

$$
\begin{array}{rclcrcl}
q(x, f(y)) & \longrightarrow & q'(f'(x), y) & \text{if} & \delta(q, f) & = & (q', f', R) \\
q(g(x), f(y)) & \longrightarrow & q'(x, g(f'(y))) & \text{if} & \delta(q, f) & = & (q', f', L)
\end{array}
$$

And special rewrite rules for dealing with the left-/rightmost blank:

$$
\begin{array}{rclcrcl}
q(\triangleright, f(y)) & \longrightarrow & q'(\triangleright, \Box(f'(y))) & \text{if} & \delta(q, f) & = & (q', f', L) \\
q(x, \triangleright) & \longrightarrow & q'(f'(x), \triangleright) & \text{if} & \delta(q, \Box) & = & (q', f', R) \\
q(g(x), \triangleright) & \longrightarrow & q'(x, g(f'(\triangleright))) & \text{if} & \delta(q, \Box) & = & (q', f', L) \\
q(\triangleright, \triangleright) & \longrightarrow & q'(\triangleright, \Box(f'(\triangleright))) & \text{if} & \delta(q, \Box) & = & (q', f', L)
\end{array}
$$

Σ^0_1-completeness of WCR

WCR is in Σ^0_1: By the Critical Pairs Lemma, WCR_R holds if and only if all critical pairs of R are convergent.
A Turing machine can compute on the input of a TRS R all (finitely many) critical pairs, and on the input of a TRS R and a term t all (finitely many) one step reducts of t.

Σ^0_1-completeness of WCR

WCR is in Σ^0_1: By the Critical Pairs Lemma, WCR_R holds if and only if all critical pairs of R are convergent.
A Turing machine can compute on the input of a TRS R all (finitely many) critical pairs, and on the input of a TRS R and a term t all (finitely many) one step reducts of t.
So it suffices to show that the following is in Σ^0_1:

> *Decide on the input of a TRS S, $n \in \mathbb{N}$ and terms $t_1, s_1, \ldots, t_n, s_n$ whether for every $i = 1, \ldots, n$ the terms t_i and s_i have a common reduct.*

This property can easily be described by a Σ^0_1 formula.

Σ_1^0-completeness of WCR

WCR is Σ_1^0-hard: We define TRS S to consist of the rules of R_{M} extended by the following:

$$\mathsf{run} \to \mathsf{T} \qquad \mathsf{run} \to q_0(\triangleright, \triangleright)$$
$$q(x, f(y)) \to \mathsf{T} \quad \text{for every } f \in \Gamma \text{ such that } \delta(q, f) \text{ is undefined} .$$

The only critical pair is $\mathsf{T} \leftarrow \mathsf{run} \to q_0(\triangleright, \triangleright)$. We have:

$$q_0(\triangleright, \triangleright) \to_S^* \mathsf{T} \text{ if and only if M halts on the blank tape.}$$

So:

$$\mathsf{WCR}(S) \text{ if and only if M halts on the blank tape.}$$

Π^0_2-completeness of CR

CR is in Π^0_2:

$$
\begin{aligned}
\mathrm{CR}_R \iff\ & \forall t \in \mathbb{N}.\ \forall r_1, r_2 \in \mathbb{N}.\ \exists r_1', r_2' \in \mathbb{N}. \\
& (((t \text{ is a term}) \text{ and } (r_1,\ r_2 \text{ are reductions}) \\
& \quad \text{and } t \equiv \mathit{first}(r_1) \equiv \mathit{first}(r_2)) \\
& \quad \Rightarrow ((r_1' \text{ and } r_2' \text{ are reductions}) \\
& \qquad \text{and } (\mathit{last}(r_1) \equiv \mathit{first}(r_1')) \text{ and } (\mathit{last}(r_2) \equiv \mathit{first}(r_2'))\ . \\
& \qquad \text{and } (\mathit{last}(r_1') \equiv \mathit{last}(r_2'))))
\end{aligned}
$$

Π_2^0-hardness of CR

We change the TRS R_M in such a way that

$$M \text{ halts on all inputs} \iff R_M \text{ is CR}$$

Π^0_2-hardness of CR

We change the TRS R_M in such a way that

$$M \text{ halts on all inputs } \iff R_M \text{ is CR}$$

Idea: use an extension of R_M with the following rules:

$$\begin{aligned} \mathsf{run}(x,y) &\to \mathsf{T} \\ \mathsf{run}(x,y) &\to q_0(x,y) \\ q(x,f(y)) &\to \mathsf{T} \qquad\qquad \text{for every } f \in \Gamma \text{ with } \delta(q,f) \text{ undefined} \end{aligned}$$

Then it seems that

$$\mathrm{CR}(R_M^+) \iff \text{ the Turing machine M halts on all configurations.}$$

Π^0_2-hardness of CR

We change the TRS R_M in such a way that

$$M \text{ halts on all inputs } \iff R_M \text{ is CR}$$

Idea: use an extension of R_M with the following rules:

$$\begin{aligned} \mathsf{run}(x, y) &\to \mathsf{T} \\ \mathsf{run}(x, y) &\to q_0(x, y) \\ q(x, f(y)) &\to \mathsf{T} \qquad \text{for every } f \in \Gamma \text{ with } \delta(q, f) \text{ undefined} \end{aligned}$$

Then it seems that

$$\mathrm{CR}(R_M^+) \iff \text{ the Turing machine M halts on all configurations.}$$

However, we only have $\Longrightarrow$. With $\Longleftarrow$ a problem arises if s and t contain variables.

Π^0_2-hardness of CR

For a Turing machines M we define the TRS S_{M} as R_M extended with

$$\mathsf{run}(x, \triangleright) \to \mathsf{T} \tag{1}$$

$$\mathsf{run}(\triangleright, y) \to q_0(\triangleright, y) \tag{2}$$

$$q(x, f(y)) \to \mathsf{T} \qquad \text{if } \delta(q, f) \text{ undefined} \tag{3}$$

$$\mathsf{run}(x, \mathsf{S}(y)) \to \mathsf{run}(\mathsf{S}(x), y) \tag{4}$$

$$\mathsf{run}(\mathsf{S}(x), y) \to \mathsf{run}(x, \mathsf{S}(y)) \,. \tag{5}$$

Π^0_2-hardness of CR

For a Turing machines M we define the TRS S_M as R_M extended with

$$\begin{aligned}
\mathsf{run}(x, \triangleright) &\to \mathsf{T} && && (1)\\
\mathsf{run}(\triangleright, y) &\to q_0(\triangleright, y) && && (2)\\
q(x, f(y)) &\to \mathsf{T} && \text{if } \delta(q, f) \text{ undefined} && (3)\\
\mathsf{run}(x, \mathsf{S}(y)) &\to \mathsf{run}(\mathsf{S}(x), y) && && (4)\\
\mathsf{run}(\mathsf{S}(x), y) &\to \mathsf{run}(x, \mathsf{S}(y))\,. && && (5)
\end{aligned}$$

Then the only cause for non-confluence can be (t_1, t_2 are ground terms)

$$q_0(\triangleright, s_1) \leftarrow_{(2)} \mathsf{run}(s_1, \triangleright) \leftarrow^*_{(4)} \mathsf{run}(t_1, t_2) \to^*_{(5)} \mathsf{run}(s_1, \triangleright) \to_{(1)} \mathsf{T}$$

Thus we can prove

$$\mathsf{CR}(S_M) \iff \text{the Turing machine M halts on all inputs.}$$

Dependency Pair problems for TRSs

- For relations $\rightarrow_R, \rightarrow_S$ we write $\rightarrow_R / \rightarrow_S$ for $\rightarrow_S^* \cdot \rightarrow_R$.
- $\rightarrow_{R,\epsilon}$ denotes R-reduction,but only at the top of a term.
- Write $\mathrm{SN}(R_{\mathrm{top}}/S)$ instead of $\mathrm{SN}(\rightarrow_{R,\epsilon}/\rightarrow_S)$.

Dependency Pair problems for TRSs

- For relations $\rightarrow_R, \rightarrow_S$ we write $\rightarrow_R / \rightarrow_S$ for $\rightarrow_S^* \cdot \rightarrow_R$.
- $\rightarrow_{R,\epsilon}$ denotes R-reduction,but only at the top of a term.
- Write $\mathrm{SN}(R_{\mathrm{top}}/S)$ instead of $\mathrm{SN}(\rightarrow_{R,\epsilon}/\rightarrow_S)$.

$\mathrm{SN}(R_{\mathrm{top}}/S)$ is the finiteness of the dependency pair problem for $\{R, S\}$.
So $\mathrm{SN}(R_{\mathrm{top}}/S)$ means that every infinite $\rightarrow_{R,\epsilon} \cup \rightarrow_S$ reduction, contains only finitely many $\rightarrow_{R,\epsilon}$ steps.

Dependency Pair problems for TRSs

- For relations $\to_R, \to_S$ we write $\to_R / \to_S$ for $\to_S^* \cdot \to_R$.
- $\to_{R,\epsilon}$ denotes R-reduction,but only at the top of a term.
- Write $\mathrm{SN}(R_{\mathrm{top}}/S)$ instead of $\mathrm{SN}(\to_{R,\epsilon}/\to_S)$.

$\mathrm{SN}(R_{\mathrm{top}}/S)$ is the finiteness of the dependency pair problem for $\{R, S\}$.
So $\mathrm{SN}(R_{\mathrm{top}}/S)$ means that every infinite $\to_{R,\epsilon} \cup \to_S$ reduction, contains only finitely many $\to_{R,\epsilon}$ steps.
Motivation: There a simple syntactic construction DP such that for any TRS S we have

$$\mathrm{SN}(\mathrm{DP}(S)_{\mathrm{top}}/S) \iff \mathrm{SN}(S).$$

Dependency pair problems

The dependency pair problem $\{R, S\}$ is finite if $\mathsf{SN}(R_{\mathsf{top}}/S)$.

$$\mathsf{SN}(R_{\mathsf{top}}/S) \;:=\; \to_S^* \cdot \to_{R,\epsilon} \text{ is SN}$$

This seems a "standard" SN-for-TRS problem, so should be Π_2^0 ...

Dependency pair problems

The dependency pair problem $\{R, S\}$ is finite if $\mathsf{SN}(R_{\mathsf{top}}/S)$.

$$\mathsf{SN}(R_{\mathsf{top}}/S) \;:=\; \rightarrow_S^* \cdot \rightarrow_{R,\epsilon} \;\text{ is SN}$$

This seems a "standard" SN-for-TRS problem, so should be Π_2^0 ...
But: $\rightarrow_S^* \cdot \rightarrow_{R,\epsilon}$ is not finitely branching.

Example

$f(x) \longrightarrow_S g(f(x))$
$g(x) \longrightarrow_R a$
Finite DP problem, but $\rightarrow_S^* \cdot \rightarrow_{R,\epsilon}$ is not finitely branching:
$f(x) \rightarrow_S^* g^n((f(x)) \rightarrow_{R,\epsilon} a$.

SN for non-finitely branching systems (ARSs)

$$\mathsf{SN}_R(a) := \forall \alpha : \mathbb{N} \to \mathbb{N}\, (\alpha(0) = a \implies \exists i\, \neg(\alpha(i) \longrightarrow_R \alpha(i+1)))$$

"There is no infinite reduction starting from *a*".
This is a Π^1_1-statement, so finiteness of DP problems is in the class Π^1_1.

SN for non-finitely branching systems (ARSs)

$$\mathsf{SN}_R(a) := \forall \alpha : \mathbb{N}\to\mathbb{N}\,(\alpha(0) = a \Longrightarrow \exists i\, \neg(\alpha(i) \longrightarrow_R \alpha(i+1)))$$

"There is no infinite reduction starting from a".
This is a Π^1_1-statement, so finiteness of DP problems is in the class Π^1_1.
Is it Π^1_1-complete?
Yes: we prove

$$\mathsf{WF}(>_M) \Longleftrightarrow \mathsf{SN}(S^M_{\mathsf{top}}/S^M)$$

for a suitable S_M constructed from M. This reduces $\mathsf{WF}(>_M)$ to $\mathsf{SN}(S^M_{\mathsf{top}}/S^M)$, thus showing Π^1_1-hardness of dependency pair problems.

DP is Π^1_1-complete

We now reduce well-foundedness of $>_M$ to $\mathrm{SN}(S^M_{\mathrm{top}}/S^M)$ and thus obtain that DP is Π^1_1-complete.

DP is Π^1_1-complete

We now reduce well-foundedness of $>_M$ to $\mathrm{SN}(S^M_{\mathrm{top}}/S^M)$ and thus obtain that DP is Π^1_1-complete.

IDEA: We define a TRS S^M such that

S^M has an infinite reduction iff $\neg\mathrm{WF}(>_M)$,

and this reduction "keeps coming back to the top level".

DP is Π^1_1-complete

We now reduce well-foundedness of $>_M$ to $\mathrm{SN}(S^M_{\mathrm{top}}/S^M)$ and thus obtain that DP is Π^1_1-complete.
IDEA: We define a TRS S^M such that
S^M has an infinite reduction iff $\neg\mathrm{WF}(>_M)$,
and this reduction "keeps coming back to the top level".
We want to mimick a computation that
1. arbitrarily picks a number n_1
2. arbitrarily picks a number n_2
3. checks if $n_1 >_M n_2$, if "no" stops, if "yes" replaces n_1 by n_2 and continues with (2)

Notation: we write $\overline{n}$ to denote $S^n(0(\triangleright))$

DP is Π^1_1-complete

First we add

$$q(x, 0(y)) \quad \longrightarrow \quad \mathsf{T} \quad \text{if} \quad \delta(q, 0) = \text{undefined}$$

so that we have

$$n >_M p \text{ iff } q_0(\overline{n}, \overline{p}) \rightarrow^*_R \mathsf{T}$$

DP is Π^1_1-complete

First we add

$$q(x, 0(y)) \longrightarrow \mathsf{T} \quad \text{if} \quad \delta(q, 0) = \text{undefined}$$

so that we have

$$n >_M p \text{ iff } q_0(\overline{n}, \overline{p}) \rightarrow^*_R \mathsf{T}$$

Then (but this is too simple ...): to pick arbitrary numbers we introduce the following TRS

$$\begin{aligned} \text{pick} &\longrightarrow \mathsf{S}(\text{pick}) \\ \text{pick} &\longrightarrow 0(\triangleright) \end{aligned}$$

and we add

$$\text{try}(\mathsf{T}, x, y) \longrightarrow \text{try}(q(x, y), y, \text{pick})$$

DP is Π^1_1-complete

The intention is to have

$$\begin{array}{rll} \mathsf{try}(\mathsf{T}, \mathsf{pick}, \mathsf{pick}) & \to^* & \\ \mathsf{try}(\mathsf{T}, \overline{n}_1, \overline{n}_2) & \to^* & \mathsf{try}(q_0(\overline{n}_1, \overline{n}_2), \overline{n}_2, \mathsf{pick}) \longrightarrow \\ \mathsf{try}(\mathsf{T}, \overline{n}_2, \overline{n}_3) & \to^* & \mathsf{try}(q_0(\overline{n}_2, \overline{n}_3), \overline{n}_3, \mathsf{pick}) \longrightarrow \ldots \end{array}$$

only if there is an infinite descending sequence $n_1 >_M n_2 >_M n_3 \ldots$

DP is Π^1_1-complete

The intention is to have

$$\begin{aligned}
\mathrm{try}(\mathsf{T}, \mathrm{pick}, \mathrm{pick}) &\rightarrow^* \\
\mathrm{try}(\mathsf{T}, \overline{n}_1, \overline{n}_2) &\rightarrow^* \quad \mathrm{try}(q_0(\overline{n}_1, \overline{n}_2), \overline{n}_2, \mathrm{pick}) \longrightarrow \\
\mathrm{try}(\mathsf{T}, \overline{n}_2, \overline{n}_3) &\rightarrow^* \quad \mathrm{try}(q_0(\overline{n}_2, \overline{n}_3), \overline{n}_3, \mathrm{pick}) \longrightarrow \dots
\end{aligned}$$

only if there is an infinite descending sequence $n_1 >_M n_2 >_M n_3 \dots$
However we also have:

$$\begin{aligned}
\mathrm{try}(\mathsf{T}, \mathrm{pick}, \mathrm{pick}) &\rightarrow^* \\
\mathrm{try}(q_0(\mathrm{pick}, \mathrm{pick}), \mathrm{pick}, \mathrm{pick}) &\rightarrow^* \quad \mathrm{try}(q_0(\overline{n}_1, \overline{n}_2), \mathrm{pick}, \mathrm{pick}) \rightarrow^* \\
\mathrm{try}(\mathsf{T}, \mathrm{pick}, \mathrm{pick}) &\rightarrow^* \quad \dots
\end{aligned}$$

if $n_1 >_M n_2$

DP is Π^1_1-complete

The intention is to have

$$\begin{array}{rll} \mathrm{try}(\mathsf{T}, \mathrm{pick}, \mathrm{pick}) & \to^* & \\ \mathrm{try}(\mathsf{T}, \overline{n}_1, \overline{n}_2) & \to^* & \mathrm{try}(q_0(\overline{n}_1, \overline{n}_2), \overline{n}_2, \mathrm{pick}) \longrightarrow \\ \mathrm{try}(\mathsf{T}, \overline{n}_2, \overline{n}_3) & \to^* & \mathrm{try}(q_0(\overline{n}_2, \overline{n}_3), \overline{n}_3, \mathrm{pick}) \longrightarrow \ldots \end{array}$$

only if there is an infinite descending sequence $n_1 >_M n_2 >_M n_3 \ldots$
However we also have:

$$\begin{array}{rll} \mathrm{try}(\mathsf{T}, \mathrm{pick}, \mathrm{pick}) & \to^* & \\ \mathrm{try}(q_0(\mathrm{pick}, \mathrm{pick}), \mathrm{pick}, \mathrm{pick}) & \to^* & \mathrm{try}(q_0(\overline{n}_1, \overline{n}_2), \mathrm{pick}, \mathrm{pick}) \to^* \\ \mathrm{try}(\mathsf{T}, \mathrm{pick}, \mathrm{pick}) & \to^* & \ldots \end{array}$$

if $n_1 >_M n_2$
Problem: $\mathrm{try}(\mathsf{T}, u, s)$ should only reduce if u and s represent a number.

DP is Π^1_1-complete

To pick arbitrary numbers we introduce the following TRS

$$\begin{array}{rcl} \mathsf{pick} & \longrightarrow & c(\mathsf{pick}) \\ \mathsf{pick} & \longrightarrow & \mathsf{ok}(0(\triangleright)) \\ c(\mathsf{ok}(x)) & \longrightarrow & \mathsf{ok}(\mathsf{S}(x)) \end{array}$$

Then $\mathsf{pick} \to^* c^n(\mathsf{pick}) \longrightarrow c^n(\mathsf{ok}(0(\triangleright))) \longrightarrow \mathsf{ok}(\mathsf{S}^n(0(\triangleright))) \equiv \overline{n}$

DP is Π^1_1-complete

To pick arbitrary numbers we introduce the following TRS

$$\begin{array}{rcl} \mathsf{pick} & \longrightarrow & c(\mathsf{pick}) \\ \mathsf{pick} & \longrightarrow & \mathsf{ok}(0(\triangleright)) \\ c(\mathsf{ok}(x)) & \longrightarrow & \mathsf{ok}(\mathsf{S}(x)) \end{array}$$

Then $\mathsf{pick} \rightarrow^* c^n(\mathsf{pick}) \longrightarrow c^n(\mathsf{ok}(0(\triangleright))) \longrightarrow \mathsf{ok}(\mathsf{S}^n(0(\triangleright))) \equiv \overline{n}$

Lemma $\mathsf{pick} \rightarrow^* \mathsf{ok}(t) \iff \exists n(t = \mathsf{S}^n(0(\triangleright)))$

DP is Π^1_1-complete

Finally we add the following rewrite rule

$$\mathsf{try}(\mathsf{T}, \mathsf{ok}(x), \mathsf{ok}(y)) \longrightarrow \mathsf{try}(q_0(x, y), \mathsf{ok}(y), \mathsf{pick})$$

Then: the term $\mathsf{try}(\mathsf{T}, \mathsf{pick}, \mathsf{pick})$ is $\mathsf{SN}(R_{\mathrm{top}}/S)$ iff $>_M$ is well-founded.

Proof: The only infinite reduction that is possible is of the form

$$\begin{array}{rcl} \mathsf{try}(\mathsf{T}, \mathsf{pick}, \mathsf{pick}) & \to^* & \\ \mathsf{try}(\mathsf{T}, \mathsf{ok}(\overline{n}_1), \mathsf{ok}(\overline{n}_2)) & \to^* & \mathsf{try}(q_0(\overline{n}_1, \overline{n}_2), \mathsf{ok}(\overline{n}_2), \mathsf{pick}) \longrightarrow \\ \mathsf{try}(\mathsf{T}, \mathsf{ok}(\overline{n}_2), \mathsf{ok}(\overline{n}_3)) & \to^* & \mathsf{try}(q_0(\overline{n}_2, \overline{n}_3), \mathsf{ok}(\overline{n}_3), \mathsf{pick}) \longrightarrow \\ & \ldots & \end{array}$$

if $n_1 >_M n_2 >_M n_3 \ldots$

Remarks / Conclusions / Future work

Remarks

- In $\mathrm{DP}^{\mathrm{min}}$, we restrict $\to_S^* \cdot \to_{R,\epsilon}$ to terms that are $\mathrm{SN}(S)$. $\mathrm{DP}^{\mathrm{min}}$ is Π_2^0-complete (see paper).
- $\mathrm{SN}^\omega(R)$ is Π_1^1-complete (see paper).

Future work:

- Characterize "all" properties of TRSs, distinguishing between "ground terms" and "all terms": UN, ...
- Characterize $\mathrm{WN}^\omega(R)$.
  $\mathrm{WN}^\omega(R) := \forall t \exists \alpha (\ldots) \iff \exists \alpha \forall t (\ldots) \in \Pi_1^1$.
- Extend to infinite terms. $\mathrm{SN}_\infty^\omega(R) := \forall \beta \forall \alpha (\ldots) \in \Pi_1^1$.
  $\mathrm{WN}_\infty^\omega(R) := \forall \beta \exists \alpha (\ldots) \in \Sigma_2^1$.
- Make the generalizations to SN^∞, WN^∞ precise, for reduction of all countable ordinal length.
- Study the proof-theoretic complexity of productivity