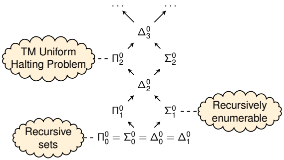
Part 6: Complexity of Productivity

Jörg Endrullis Clemens Grabmayer Dimitri Hendriks

Vrije Universiteit Amsterdam – Universiteit Utrecht – Vrije Universiteit Amsterdam



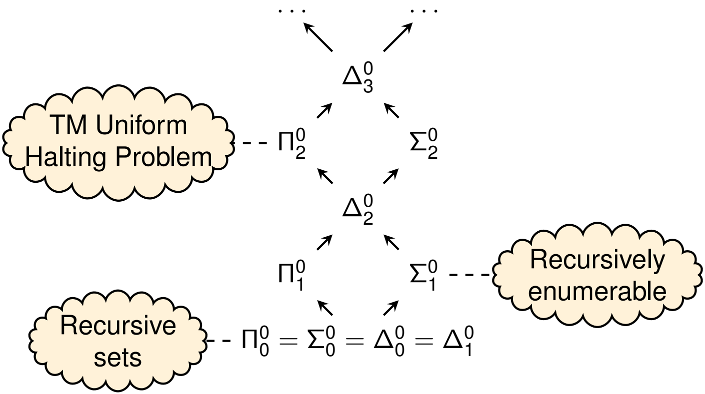
ISR 2010, Utrecht University

July 8, 2010

Overview

Overview

The arithmetical hierarchy

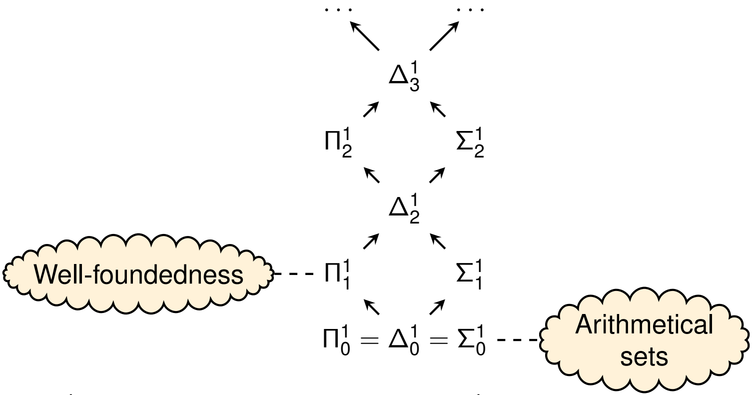


$\boldsymbol{\Pi^0_0} := \boldsymbol{\Sigma^0_0} :=$ 1st-order arithmetic formulas with bounded quantifiers

$$\boldsymbol{\Sigma^0_{n+1}} := \{\exists x_1 \ldots \exists x_k \Psi \mid \Psi \in \boldsymbol{\Pi^0_n}\}$$
$$\boldsymbol{\Pi^0_{n+1}} := \{\forall x_1 \ldots \forall x_k \Psi \mid \Psi \in \boldsymbol{\Sigma^0_n}\}$$

$\Sigma^0_n(\Pi^0_n) :=$ interpretations of formulas in $\boldsymbol{\Sigma^0_n}(\boldsymbol{\Pi^0_n})$ over $\mathbb{N}$ $\qquad \Delta^0_n := \Sigma^0_n \cap \Pi^0_n$

The analytical hierarchy



$\boldsymbol{\Pi^1_0} := \boldsymbol{\Sigma^1_0} :=$ 2nd-order arithm. formulas without set quantifiers

$\boldsymbol{\Sigma^1_{n+1}} := \{\exists X_1 \ldots \exists X_k \Psi \mid \Psi \in \boldsymbol{\Pi^1_n}\}$

$\boldsymbol{\Pi^1_{n+1}} := \{\forall X_1 \ldots \forall X_k \Psi \mid \Psi \in \boldsymbol{\Sigma^1_n}\}$

$\Sigma^1_n(\Pi^1_n) :=$ interpretations of formulas in $\boldsymbol{\Sigma^1_n}(\boldsymbol{\Pi^1_n})$ over $\mathbb{N}$ $\quad \Delta^1_n := \Sigma^1_n \cap \Pi^1_n$

Overview

1. The arithmetical and analytical hierarchies
2. Complexity of productivity and equivalence for stream spec's
3. Productivity and variant definitions in TRSs
4. Complexity of productivity, and variants, in TRSs
5. Summary and References

Productivity and equivalence problems

PRODUCTIVITY PROBLEM for class $\mathcal{C}$ of stream spec's

Instance: A stream specification $\mathcal{R} \in \mathcal{C}$ with root M_0

Question: Is $\mathcal{R}$ productive?

(Does $\mathsf{M}_0 \twoheadrightarrow u_0 : u_1 : u_2 : u_3 : \ldots$?)

EQUIVALENCE PROBLEM for class $\mathcal{C}$ of stream spec's

Instance: Stream specifications $\mathcal{R}_1, \mathcal{R}_2 \in \mathcal{C}$ with roots $\mathsf{M}_0^{(1)}, \mathsf{M}_0^{(2)}$

Question: Do $\mathcal{R}_1$ and $\mathcal{R}_2$ uniquely define the same stream? *

*) E.g. in the case that $\mathsf{M}_0^{(1)} \twoheadrightarrow u_0 : u_1 : u_2 : u_3 : \ldots \twoheadleftarrow \mathsf{M}_0^{(2)}$.

Productivity and equivalence problems

PRODUCTIVITY PROBLEM for class $\mathcal{C}$ of stream spec's

Instance: A stream specification $\mathcal{R} \in \mathcal{C}$ with root M_0

Question: Is $\mathcal{R}$ productive?

(Does $\mathsf{M}_0 \twoheadrightarrow u_0 : u_1 : u_2 : u_3 : \ldots$?)

EQUIVALENCE PROBLEM for class $\mathcal{C}$ of stream spec's

Instance: Stream specifications $\mathcal{R}_1, \mathcal{R}_2 \in \mathcal{C}$ with roots $\mathsf{M}_0^{(1)}, \mathsf{M}_0^{(2)}$

Question: Do $\mathcal{R}_1$ and $\mathcal{R}_2$ uniquely define the same stream? *

*) E.g. in the case that $\mathsf{M}_0^{(1)} \twoheadrightarrow u_0 : u_1 : u_2 : u_3 : \ldots \twoheadleftarrow \mathsf{M}_0^{(2)}$.

Productivity and equivalence problems

PRODUCTIVITY PROBLEM for class $\mathcal{C}$ of stream spec's

Instance: A stream specification $\mathcal{R} \in \mathcal{C}$ with root M_0

Question: Is $\mathcal{R}$ productive?

(Does $\mathsf{M}_0 \twoheadrightarrow u_0 : u_1 : u_2 : u_3 : \ldots$?)

EQUIVALENCE PROBLEM for class $\mathcal{C}$ of stream spec's

Instance: Stream specifications $\mathcal{R}_1, \mathcal{R}_2 \in \mathcal{C}$ with roots $\mathsf{M}_0^{(1)}, \mathsf{M}_0^{(2)}$

Question: Do $\mathcal{R}_1$ and $\mathcal{R}_2$ uniquely define the same stream? ⋆

⋆) E.g. in the case that $\mathsf{M}_0^{(1)} \twoheadrightarrow u_0 : u_1 : u_2 : u_3 : \ldots \twoheadleftarrow \mathsf{M}_0^{(2)}$.

Complexity of productivity and equivalence

Equivalence problem for:

- automatic sequences: (easily) decidable
- morphic streams: decidable [Culik and Harju (1984)]

stream specification	productivity probl.	equivalence probl.
productive	—	Π^0_1-complete
pure and pure$^+$	decidable	Π^0_1-hard
flat	Π^0_2-complete	Π^0_2-complete
general	Π^0_2-complete†	Π^0_2-complete*

*) G. Roşu (2006).

†) J. Grue Simonsen (2009), E/G/H (2009).

Complexity of productivity and equivalence

Equivalence problem for:

- automatic sequences: (easily) decidable
- morphic streams: decidable [Culik and Harju (1984)]

stream specification	productivity probl.	equivalence probl.
productive	—	Π^0_1-complete
pure and pure$^+$	decidable	Π^0_1-hard
flat	Π^0_2-complete	Π^0_2-complete
general	Π^0_2-complete†	Π^0_2-complete*

*) G. Roşu (2006).

†) J. Grue Simonsen (2009), E/G/H (2009).

Complexity of productivity and equivalence

Equivalence problem for:

- automatic sequences: (easily) decidable
- morphic streams: decidable [Culik and Harju (1984)]

stream specification	productivity probl.	equivalence probl.
productive	—	Π^0_1-complete
pure and pure$^+$	decidable	Π^0_1-hard
flat	Π^0_2-complete	Π^0_2-complete
general	Π^0_2-complete†	Π^0_2-complete*

$\star$) G. Roşu (2006).

$\dagger$) J. Grue Simonsen (2009), E/G/H (2009).

Complexity of productivity and equivalence

Equivalence problem for:

- automatic sequences: (easily) decidable
- morphic streams: decidable [Culik and Harju (1984)]

stream specification	productivity probl.	equivalence probl.
productive	—	Π^0_1-complete
pure and pure$^+$	decidable	Π^0_1-hard
flat	Π^0_2-complete	Π^0_2-complete
general	Π^0_2-complete†	Π^0_2-complete*

*) G. Roşu (2006).

†) J. Grue Simonsen (2009), E/G/H (2009).

Complexity of productivity and equivalence

Equivalence problem for:

- automatic sequences: (easily) decidable
- morphic streams: decidable [Culik and Harju (1984)]

stream specification	productivity probl.	equivalence probl.
productive	—	Π^0_1-complete
pure and pure$^+$	decidable	Π^0_1-hard
flat	Π^0_2-complete	Π^0_2-complete
general	Π^0_2-complete†	Π^0_2-complete*

*) G. Roşu (2006).

†) J. Grue Simonsen (2009), E/G/H (2009).

Complexity of productivity and equivalence

Equivalence problem for:

- automatic sequences: (easily) decidable
- morphic streams: decidable [Culik and Harju (1984)]

stream specification	productivity probl.	equivalence probl.
productive	—	Π^0_1-complete
pure and pure$^+$	decidable	Π^0_1-hard
flat	Π^0_2-complete	Π^0_2-complete
general	Π^0_2-complete†	Π^0_2-complete*

*) G. Roşu (2006).

†) J. Grue Simonsen (2009), E/G/H (2009).

Complexity of productivity and equivalence

Equivalence problem for:

- automatic sequences: (easily) decidable
- morphic streams: decidable [Culik and Harju (1984)]

stream specification	productivity probl.	equivalence probl.
productive	—	Π^0_1-complete
pure and pure$^+$	decidable	Π^0_1-hard
flat	Π^0_2-complete	Π^0_2-complete
general	Π^0_2-complete†	Π^0_2-complete*

*) G. Roşu (2006).

†) J. Grue Simonsen (2009), E/G/H (2009).

Complexity of productivity and equivalence

Equivalence problem for:

- automatic sequences: (easily) decidable
- morphic streams: decidable [Culik and Harju (1984)]

stream specification	**productivity probl.**	**equivalence probl.**
productive	—	Π^0_1-complete
pure and pure$^+$	decidable	Π^0_1-hard
flat	Π^0_2-complete	Π^0_2-complete
general	Π^0_2-complete†	Π^0_2-complete*

*) G. Roşu (2006).

†) J. Grue Simonsen (2009), E/G/H (2009).

Productivity of flat stream specifications

Theorem

The productivity problem for flat stream specifications is Π^0_2-complete.

Proof.

Contained in Π^0_2:

A flat stream spec $\mathcal{R}$ with root M_0 is productive iff

$$M_0 \twoheadrightarrow u_0 : u_1 : u_2 : \ldots,$$

and iff:

$$\left.\begin{array}{l}\forall n \in \mathbb{N}.\ \exists m \in \mathbb{N}.\ \exists \rho.\ \ \rho \text{ is rewrite sequence of length } m, \\ \qquad\qquad\qquad\qquad\quad\ \rho : M_0 \twoheadrightarrow u_0 : u_1 : u_2 : \ldots u_n : t\end{array}\right\} \in \Pi^0_2$$

Π^0_2-complete: By reducing the uniform halting problem for Turing-machines, which is Π^0_2-complete, to the productivity problem.

Productivity of flat stream specifications

Theorem

The productivity problem for flat stream specifications is Π^0_2-complete.

Proof.

Contained in Π^0_2:

A flat stream spec $\mathcal{R}$ with root M_0 is productive iff

$$\mathsf{M}_0 \twoheadrightarrow u_0 : u_1 : u_2 : \ldots,$$

and iff:

$$\left.\forall n \in \mathbb{N}.\ \exists m \in \mathbb{N}.\ \exists \rho.\ \begin{array}{l} \rho \text{ is rewrite sequence of length } m, \\ \rho : \mathsf{M}_0 \twoheadrightarrow u_0 : u_1 : u_2 : \ldots u_n : t \end{array}\right\} \in \mathbf{\Pi^0_2}$$

Π^0_2-complete: By reducing the uniform halting problem for Turing-machines, which is Π^0_2-complete, to the productivity problem.

Productivity of flat stream specifications

Theorem

The productivity problem for flat stream specifications is Π^0_2-complete.

Proof.

Contained in Π^0_2:

A flat stream spec $\mathcal{R}$ with root M_0 is productive iff

$$\mathsf{M}_0 \twoheadrightarrow u_0 : u_1 : u_2 : \ldots,$$

and iff:

$$\left.\begin{array}{l} \forall n \in \mathbb{N}.\ \exists m \in \mathbb{N}.\ \exists \rho.\ \ \rho \text{ is rewrite sequence of length } m, \\ \qquad\qquad\qquad\qquad\quad\ \rho : \mathsf{M}_0 \twoheadrightarrow u_0 : u_1 : u_2 : \ldots u_n : t \end{array}\right\} \in \mathbf{\Pi^0_2}$$

Π^0_2-complete: By reducing the uniform halting problem for Turing-machines, which is Π^0_2-complete, to the productivity problem.

Productivity of flat stream specifications

Proof (continued).

We show $\{\ulcorner M \urcorner : M \text{ halts on all inputs}\} = \mathit{UHP} \leq_m \mathit{PROD}(\mathit{FLAT})$:

An instance $\ulcorner M \urcorner$ of UHP is transformed into the flat spec $\mathcal{R}_M$:

$$\mathsf{R}_M \to \mathsf{R}(\mathsf{stops}_M(\mathsf{0}, \mathsf{0}), \mathsf{0}, \mathsf{0})$$

$$\mathsf{R}(\mathsf{s}(\mathsf{0}), x, y) \to \mathsf{R}(\mathsf{stops}_M(x, \mathsf{s}(y)), x, \mathsf{s}(y))$$

$$\mathsf{R}(\mathsf{0}, x, y) \to \mathsf{0} : \mathsf{R}(\mathsf{stops}_M(\mathsf{s}(x), \mathsf{0}), \mathsf{s}(x), \mathsf{0})$$

$$\mathsf{stops}_M(x, y) \twoheadrightarrow \begin{cases} \mathsf{0} \ldots & M \text{ halts on } x \text{ in } \leq y \text{ steps} \\ \mathsf{s}(\mathsf{0}) \ldots & \text{otherwise} \end{cases}$$

Then: $\mathcal{R}_M$ is productive (and: $\mathsf{R}_M \twoheadrightarrow \mathsf{0} : \mathsf{0} : \ldots$)

$\iff$ M halts on all inputs

$\iff$ $\ulcorner M \urcorner \in \mathit{UHP}$.

Productivity of flat stream specifications

Proof (continued).

We show $\{\ulcorner M \urcorner : M \text{ halts on all inputs}\} = UHP \leq_m PROD(FLAT)$:
An instance $\ulcorner M \urcorner$ of UHP is transformed into the flat spec $\mathcal{R}_M$:

$$\mathsf{R}_M \to \mathsf{R}(\mathsf{stops}_M(0,0),0,0)$$

$$\mathsf{R}(\mathsf{s}(0),x,y) \to \mathsf{R}(\mathsf{stops}_M(x,\mathsf{s}(y)),x,\mathsf{s}(y))$$

$$\mathsf{R}(0,x,y) \to 0 : \mathsf{R}(\mathsf{stops}_M(\mathsf{s}(x),0),\mathsf{s}(x),0)$$

$$\mathsf{stops}_M(x,y) \twoheadrightarrow \begin{cases} 0 \ldots & M \text{ halts on } x \text{ in} \leq y \text{ steps} \\ \mathsf{s}(0) \ldots & \text{otherwise} \end{cases}$$

Then: $\mathcal{R}_M$ is productive (and: $\mathsf{R}_M \twoheadrightarrow^\omega 0 : 0 : \ldots$)

$\iff M$ halts on all inputs

$\iff \ulcorner M \urcorner \in UHP$.

Productivity of flat stream specifications

Proof (continued).

We show $\{\ulcorner M \urcorner : M \text{ halts on all inputs}\} = \mathit{UHP} \leq_m \mathit{PROD}(\mathit{FLAT})$:
An instance $\ulcorner M \urcorner$ of UHP is transformed into the flat spec $\mathcal{R}_M$:

$$\mathsf{R}_M \to \mathsf{R}(\mathsf{stops}_M(0,0),0,0)$$

$$\mathsf{R}(\mathsf{s}(0),x,y) \to \mathsf{R}(\mathsf{stops}_M(x,\mathsf{s}(y)),x,\mathsf{s}(y))$$

$$\mathsf{R}(0,x,y) \to 0 : \mathsf{R}(\mathsf{stops}_M(\mathsf{s}(x),0),\mathsf{s}(x),0)$$

$$\mathsf{stops}_M(x,y) \twoheadrightarrow \begin{cases} 0 \ldots & M \text{ halts on } x \text{ in} \leq y \text{ steps} \\ \mathsf{s}(0) \ldots & \text{otherwise} \end{cases}$$

Then: $\mathcal{R}_M$ is productive (and: $\mathsf{R}_M \twoheadrightarrow\!\!\!\!\!\twoheadrightarrow 0 : 0 : \ldots$)

$\iff M$ halts on all inputs

$\iff \ulcorner M \urcorner \in \mathit{UHP}$.

Productivity of flat stream specifications

Proof (continued).

We show $\{\ulcorner M \urcorner : M \text{ halts on all inputs}\} = \mathit{UHP} \leq_m \mathit{PROD}(\mathit{FLAT})$:

An instance $\ulcorner M \urcorner$ of UHP is transformed into the flat spec $\mathcal{R}_M$:

$$\begin{aligned} \mathsf{R}_M &\to \mathsf{R}(\mathsf{stops}_M(0,0),0,0) \\ \mathsf{R}(\mathsf{s}(0),x,y) &\to \mathsf{R}(\mathsf{stops}_M(x,\mathsf{s}(y)),x,\mathsf{s}(y)) \\ \mathsf{R}(0,x,y) &\to 0 : \mathsf{R}(\mathsf{stops}_M(\mathsf{s}(x),0),\mathsf{s}(x),0) \end{aligned}$$

$$\mathsf{stops}_M(x,y) \twoheadrightarrow \begin{cases} 0 \ldots & M \text{ halts on } x \text{ in } \leq y \text{ steps} \\ \mathsf{s}(0) \ldots & \text{otherwise} \end{cases}$$

Then: $\mathcal{R}_M$ is productive (and: $\mathsf{R}_M \twoheadrightarrow 0 : 0 : \ldots$)

$\iff$ M halts on all inputs

$\iff$ $\ulcorner M \urcorner \in \mathit{UHP}$.

Productivity of flat stream specifications

Proof (continued).

We show $\{\ulcorner M \urcorner : M \text{ halts on all inputs}\} = \mathit{UHP} \leq_m \mathit{PROD}(\mathit{FLAT})$:

An instance $\ulcorner M \urcorner$ of UHP is transformed into the flat spec $\mathcal{R}_M$:

$$\mathsf{R}_M \to \mathsf{R}(\mathsf{stops}_M(0,0),0,0)$$

$$\mathsf{R}(\mathsf{s}(0),x,y) \to \mathsf{R}(\mathsf{stops}_M(x,\mathsf{s}(y)),x,\mathsf{s}(y))$$

$$\mathsf{R}(0,x,y) \to 0 : \mathsf{R}(\mathsf{stops}_M(\mathsf{s}(x),0),\mathsf{s}(x),0)$$

$$\mathsf{stops}_M(x,y) \twoheadrightarrow \begin{cases} 0 \ldots & M \text{ halts on } x \text{ in} \leq y \text{ steps} \\ \mathsf{s}(0) \ldots & \text{otherwise} \end{cases}$$

Then: $\mathcal{R}_M$ is productive (and: $\mathsf{R}_M \twoheadrightarrow 0 : 0 : \ldots$)

$\iff M$ halts on all inputs

$\iff \ulcorner M \urcorner \in \mathit{UHP}$.

Equivalence for productive stream specifications

Theorem

The equivalence problem for productive specifications is Π^0_1-complete.

Proof.

Π^0_1-complete: By reducing $\overline{HP}$, the complement of the halting problem, which is Π^0_1-complete, to the equivalence problem here.

Contained in Π^0_1:
Productive spec's $\mathcal{R}_1$ and $\mathcal{R}_2$ with roots $\mathsf{M}_0^{(1)}$, $\mathsf{M}_0^{(2)}$ are equivalent iff

$$\mathsf{M}_0^{(1)} \twoheadrightarrow\!\!\!\rightarrow u_0 : u_1 : u_2 : u_3 : \ldots \leftarrow\!\!\!\twoheadleftarrow \mathsf{M}_0^{(2)} ,$$

and iff:

$$\forall n, m \in \mathbb{N}.\ \forall \rho_1, \rho_2.\ \left.\begin{array}{l} \rho_1, \rho_2 \text{ are rewrite sequences of length } n, \\ \rho_1 : \mathsf{M}_0^{(1)} \twoheadrightarrow u'_0 : u'_1 : u'_2 : \ldots u'_m : t', \\ \rho_2 : \mathsf{M}_0^{(1)} \twoheadrightarrow u''_0 : u''_1 : u''_2 : \ldots u''_m : t'', \\ \Rightarrow\ nf(u'_0) = nf(u'_0) \wedge \ldots \wedge nf(u'_m) = nf(u'_m) \end{array}\right\} \in \Pi^0_1$$

Equivalence for productive stream specifications

Theorem

The equivalence problem for productive specifications is Π^0_1-complete.

Proof.

Π^0_1-complete: By reducing $\overline{HP}$, the complement of the halting problem, which is Π^0_1-complete, to the equivalence problem here.

Contained in Π^0_1:
Productive spec's $\mathcal{R}_1$ and $\mathcal{R}_2$ with roots $\mathsf{M}_0^{(1)}$, $\mathsf{M}_0^{(2)}$ are equivalent iff

$$\mathsf{M}_0^{(1)} \twoheadrightarrow\!\!\!\!\to u_0 : u_1 : u_2 : u_3 : \ldots \twoheadleftarrow\!\!\!\!\leftarrow \mathsf{M}_0^{(2)} \,,$$

and iff:

$$\forall n, m \in \mathbb{N}.\ \forall \rho_1, \rho_2.\ \left.\begin{array}{l} \rho_1, \rho_2 \text{ are rewrite sequences of length } n, \\ \rho_1 : \mathsf{M}_0^{(1)} \twoheadrightarrow u'_0 : u'_1 : u'_2 : \ldots u'_m : t', \\ \rho_2 : \mathsf{M}_0^{(1)} \twoheadrightarrow u''_0 : u''_1 : u''_2 : \ldots u''_m : t'', \\ \Rightarrow\ nf(u'_0) = nf(u'_0) \wedge \ldots \wedge nf(u'_m) = nf(u'_m) \end{array}\right\} \in \Pi^0_1$$

Equivalence for productive stream specifications

Theorem

The equivalence problem for productive specifications is Π^0_1-complete.

Proof.

Π^0_1-complete: By reducing $\overline{HP}$, the complement of the halting problem, which is Π^0_1-complete, to the equivalence problem here.

Contained in Π^0_1:
Productive spec's $\mathcal{R}_1$ and $\mathcal{R}_2$ with roots $\mathsf{M}_0^{(1)}$, $\mathsf{M}_0^{(2)}$ are equivalent iff

$$\mathsf{M}_0^{(1)} \twoheadrightarrow\!\!\!\!\twoheadrightarrow u_0 : u_1 : u_2 : u_3 : \ldots \twoheadleftarrow\!\!\!\!\twoheadleftarrow \mathsf{M}_0^{(2)} ,$$

and iff:

$$\left.\begin{array}{l} \forall n, m \in \mathbb{N}.\ \forall \rho_1, \rho_2.\ \ \rho_1, \rho_2 \text{ are rewrite sequences of length } n, \\ \qquad \rho_1 : \mathsf{M}_0^{(1)} \twoheadrightarrow u'_0 : u'_1 : u'_2 : \ldots u'_m : t', \\ \qquad \rho_2 : \mathsf{M}_0^{(1)} \twoheadrightarrow u''_0 : u''_1 : u''_2 : \ldots u''_m : t'', \\ \qquad \Rightarrow\ nf(u'_0) = nf(u'_0) \wedge \ldots \wedge nf(u'_m) = nf(u'_m) \end{array}\right\} \in \mathbf{\Pi^0_1}$$

Overview

1. The arithmetical and analytical hierarchies
2. Complexity of productivity and equivalence for stream spec's
3. Productivity and variant definitions in TRSs
4. Complexity of productivity, and variants, in TRSs
5. Summary and References

Productivity and variants

1 $$\mathsf{zeros} \to 0 : \mathsf{zeros}$$

- productive: there is only one maximal rewrite sequence:
  $\mathsf{zeros} \to 0 : \mathsf{zeros} \to 0 : 0 : \mathsf{zeros} \to \ldots \twoheadrightarrow 0 : 0 : 0 : \ldots$

2 $$\mathsf{zeros} \to 0 : \mathsf{id}(\mathsf{zeros}) \qquad \mathsf{id}(xs) \to xs$$

- $\mathsf{zeros} \twoheadrightarrow 0 : \mathsf{id}(0 : \mathsf{id}(0 : \mathsf{id}(\ldots)))$
- still productive, since for all max. outermost-fair rewrite sequences:
  $\mathsf{zeros} \twoheadrightarrow 0 : 0 : 0 : \ldots$

Even for well-behaved spec's (orthogonal TRSs), productivity should be based on a fair treatment of outermost redexes.

Productivity and variants

1. $$\mathsf{zeros} \to 0 : \mathsf{zeros}$$

   - productive: there is only one maximal rewrite sequence:
     $\mathsf{zeros} \to 0 : \mathsf{zeros} \to 0 : 0 : \mathsf{zeros} \to \ldots \twoheadrightarrow 0 : 0 : 0 : \ldots$

2. $$\mathsf{zeros} \to 0 : \mathsf{id}(\mathsf{zeros}) \qquad \mathsf{id}(xs) \to xs$$

   - $\mathsf{zeros} \twoheadrightarrow 0 : \mathsf{id}(0 : \mathsf{id}(0 : \mathsf{id}(\ldots)))$
   - still productive, since for all max. outermost-fair rewrite sequences:
     $\mathsf{zeros} \twoheadrightarrow 0 : 0 : 0 : \ldots$

Even for well-behaved spec's (orthogonal TRSs), productivity should be based on a fair treatment of outermost redexes.

Productivity and variants

1. $$\mathsf{zeros} \to 0 : \mathsf{zeros}$$

   - productive: there is only one maximal rewrite sequence:
     $\mathsf{zeros} \to 0 : \mathsf{zeros} \to 0 : 0 : \mathsf{zeros} \to \ldots \twoheadrightarrow 0 : 0 : 0 : \ldots$

2. $$\mathsf{zeros} \to 0 : \mathsf{id}(\mathsf{zeros}) \qquad \mathsf{id}(xs) \to xs$$

   - $\mathsf{zeros} \twoheadrightarrow 0 : \mathsf{id}(0 : \mathsf{id}(0 : \mathsf{id}(\ldots)))$
   - still productive, since for all max. outermost-fair rewrite sequences:
     $\mathsf{zeros} \twoheadrightarrow 0 : 0 : 0 : \ldots$

Even for well-behaved spec's (orthogonal TRSs), productivity should be based on a fair treatment of outermost redexes.

Productivity and variants

3. $\mathsf{maybe} \to 0 : \mathsf{maybe} \qquad \mathsf{maybe} \to \mathsf{sink} \qquad \mathsf{sink} \to \mathsf{sink}$

- productive or not, dependent on the chosen strategy
- 'weakly productive': $\mathsf{maybe} \twoheadrightarrow 0 : 0 : 0 : \ldots$
- not 'strongly productive': e.g. $\mathsf{maybe} \to \mathsf{sink} \to \mathsf{sink} \to \ldots$

4. $\mathsf{abitstream} \to 0 : \mathsf{abitstream} \qquad \mathsf{abitstream} \to 1 : \mathsf{abitstream}$

- productive independent of the strategy chosen
- 'weakly' and 'strongly productive'
- infinite normal forms not unique

Productivity and variants

3. $\mathsf{maybe} \to 0 : \mathsf{maybe} \qquad \mathsf{maybe} \to \mathsf{sink} \qquad \mathsf{sink} \to \mathsf{sink}$

   - productive or not, dependent on the chosen strategy
   - 'weakly productive': $\mathsf{maybe} \twoheadrightarrow\!\!\!\!\!\rightarrow 0 : 0 : 0 : \ldots$
   - not 'strongly productive': e.g. $\mathsf{maybe} \to \mathsf{sink} \to \mathsf{sink} \to \ldots$

4. $\mathsf{abitstream} \to 0 : \mathsf{abitstream} \qquad \mathsf{abitstream} \to 1 : \mathsf{abitstream}$

   - productive independent of the strategy chosen
   - 'weakly' and 'strongly productive'
   - infinite normal forms not unique

Definition of productivity in general TRSs

With practical purposes in mind, we think:

- For non-well-behaved spec's (non-orthogonal TRSs), productivity has to be defined relative to a given rewrite strategy.
- Strategy-independent variants (strong, weak productivity) are only of theoretical interest.
- Uniqueness of (infinite) normal form UN^{∞} should be considered to be a separate property, independent of productivity. (In orthogonal TRSs, UN^{∞} is guaranteed.)

Productivity w.r.t. computable strategies

Let $\mathcal{R}$ be a TRS.
A strategy for a rewrite relation $\to_{\mathcal{R}}$ is a relation $\rightsquigarrow \subseteq \to_{\mathcal{R}}$ with the same normal forms as $\to_{\mathcal{R}}$.

Definition

A term t is called productive w.r.t. a strategy $\rightsquigarrow$ if all maximal $\rightsquigarrow$-rewrite sequences starting from t end in a constructor normal form.

Strong and weak productivity

Definition

A term t in a TRS $\mathcal{R}$ is called

- strongly productive: all maximal outermost-fair rewrite sequences starting from t end in a constructor normal form.
- weakly productive: if there exists a rewrite sequence starting from t that ends in a constructor normal form.

Overview

Productivity w.r.t. computable strategies

PRODUCTIVITY PROBLEM w.r.t. a family $\mathcal{S}$ of computable strategies

Instance: Encodings of a finite TRS $\mathcal{R}$, a strategy $\leadsto \in \mathcal{S}(\mathcal{R})$, and a term t in $\mathcal{R}$.

Question: Is t productive w.r.t. $\leadsto$?

We say that:

- such a family $\mathcal{S}$ is admissible: if R is orthogonal, $\mathcal{S}(\mathcal{R}) \neq \emptyset$.

Theorem

For every family of admissible, computale strategies $\mathcal{S}$, the productivity problem w.r.t. $\mathcal{S}$ is Π^0_2-complete.

Corollary

In orthogonal TRSs, productivity w.r.t. lazy (outermost-fair) evaluation is Π^0_2-complete.

Productivity w.r.t. computable strategies

PRODUCTIVITY PROBLEM w.r.t. a family $\mathcal{S}$ of computable strategies

Instance: Encodings of a finite TRS $\mathcal{R}$, a strategy $\leadsto \in \mathcal{S}(\mathcal{R})$, and a term t in $\mathcal{R}$.

Question: Is t productive w.r.t. $\leadsto$?

We say that:

- such a family $\mathcal{S}$ is admissible: if R is orthogonal, $\mathcal{S}(\mathcal{R}) \neq \emptyset$.

Theorem

For every family of admissible, computale strategies $\mathcal{S}$, the productivity problem w.r.t. $\mathcal{S}$ is Π^0_2-complete.

Corollary

In orthogonal TRSs, productivity w.r.t. lazy (outermost-fair) evaluation is Π^0_2-complete.

Strong and weak productivity

Theorem

The recognition problem for

- *strong productivity is Π^1_1-complete;*
- *weak productivity is Σ^1_1-complete.*

Proof (Idea).

Π^1_1-hardness (Σ^1_1-hardness): reducing the
– recognition problem for well-founded (for non-well-founded) binary relations over $\mathbb{N}$, which is Π^1_1-complete (Σ^1_1-complete), to the
– to the recognition problem of strong (weak) productivity. $\square$

Uniqueness of infinite normal form

Theorem

The problem of recognising, for TRSs $\mathcal{R}$ and terms t in $\mathcal{R}$, whether t has a unique (finite or infinite) normal form is Π^1_1-complete.

Changes due to adding the condition uniqueness of normal form:

(i) w.r.t. family of strategies:
- uniqueness of normal forms w.r.t. $\rightsquigarrow$: Π^0_2-complete.
- uniqueness of normal forms generally: Π^1_1-complete.

(ii) strong productivity: Π^1_1-complete

(iii) weak productivity: now $(\Pi^1_1 \cup \Sigma^1_1)$-hard

Uniqueness of infinite normal form

Theorem

The problem of recognising, for TRSs $\mathcal{R}$ and terms t in $\mathcal{R}$, whether t has a unique (finite or infinite) normal form is Π^1_1-complete.

Changes due to adding the condition uniqueness of normal form:

(i) w.r.t. family of strategies:
- uniqueness of normal forms w.r.t. $\rightsquigarrow$: Π^0_2-complete.
- uniqueness of normal forms generally: Π^1_1-complete.

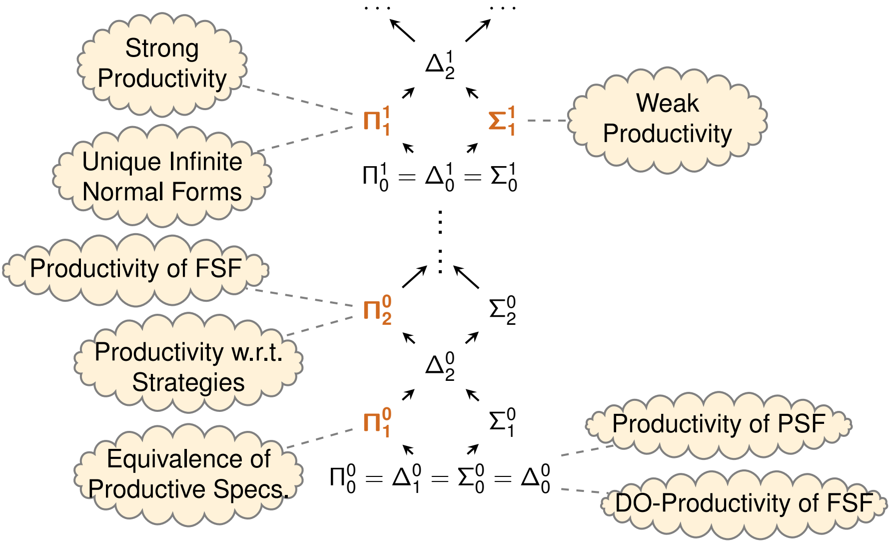
(ii) strong productivity: Π^1_1-complete

(iii) weak productivity: now $(\Pi^1_1 \cup \Sigma^1_1)$-hard

Overview

1. The arithmetical and analytical hierarchies
2. Complexity of productivity and equivalence for stream spec's
3. Productivity and variant definitions in TRSs
4. Complexity of productivity, and variants, in TRSs
5. Summary and References

Complexity of productivity: gathered results



Complexity of productivity: gathered results

- Productivity for pure/pure$^+$ stream specifications is decidable
- Productivity for flat stream specifications is Π^0_2-complete
  - But recall: data-oblivious productivity is decidable for flat spec's.
- Complexity of productivity in TRS's, and variant definitions:
  - productivity w.r.t. computable strategies: Π^0_2-complete
  - strong productivity: Π^1_1-complete
  - weak productivity: Σ^1_1-complete
  - unique infinite normal forms: Π^1_1-complete

References

Jörg Endrullis, Clemens Grabmayer, and Dimitri Hendriks.
Complexity of Fractran and Productivity.
In *CADE-22*, volume 5663 of *LNCS*, pages 371–387. Springer, 2009.

Jakob Grue Simonsen.
The Π^0_2-Completeness of Most of the Properties of Rewriting Systems You Care About (and Productivity).
In *RTA*, volume 5595 of *LNCS*, pages 335–349. Springer, 2009.

Jörg Endrullis, Herman Geuvers, and Hans Zantema.
Degrees of Undecidability of TRS Properties.
In *CSL*, volume 5771 of *LNCS*, pages 255–270. Springer, 2009.

G. Roşu.
Equality of Streams is a Π^0_2-complete Problem.
In *ICFP*, pages 184–191, 2006.