Productivity of Infinite Data Structures

Jörg Endrullis   Clemens Grabmayer   Dimitri Hendriks

Vrije Universiteit Amsterdam

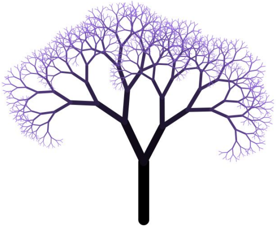
Utrecht University

ISR 2010, Utrecht University

July 8, 2010

Outline

Productivity of Trees



$T \to 0(1(T, T), T)$

```
                  0
           /             \
         1                 0
       /   \             /   \
     0       0         1       0
    / \     / \       / \     / \
   1   0   1   0     0   0   1   0
  /\  /\  /\  /\    /\  /\  /\  /\
 . ....................................... .
```

Constructor Term Rewriting

Let R be a TRS over Σ:

- defined symbols $\mathcal{D}(R) = \{\mathsf{f} \mid \mathsf{f}(\ldots) \to r \in R\}$,
- constructor symbols $\mathcal{C}(R) = \Sigma \setminus \mathcal{D}(R)$.

Definition (Constructor TRS)

…for every $\mathsf{f}(t_1, \ldots, t_n) \to r \in R$ we have $t_1, \ldots, t_n \in \mathit{Ter}(\mathcal{C}(R), \mathcal{X})$

Example (A Constructor TRS)

$$\mathsf{f}(\mathsf{s}(x), y) \to 0 \qquad\qquad \mathsf{g}(x) \to \mathsf{f}(\mathsf{f}(x, x), x)$$

Example (Not a Constructor TRS)

$$\mathsf{f}(\mathsf{g}(x), y) \to 0 \qquad\qquad \mathsf{g}(x) \to \mathsf{s}(x)$$

Exhaustivity

Definition (Exhaustive)

…every $\mathsf{f}(t_1, \ldots, t_n)$ with $\mathsf{f} \in \mathcal{D}(R)$ and $t_i \in \mathit{Ter}^\infty(\mathcal{C}(R), \varnothing)$ is a redex.

Example (A Non-Exhaustive TRS)

$$\begin{aligned} \mathsf{T} &\to \mathsf{0}(\mathsf{1}(\mathsf{h}(\mathsf{T}), \mathsf{T}), \mathsf{T}) \\ \mathsf{h}(\mathsf{0}(x, y)) &\to \mathsf{1}(x, \mathsf{h}(y)) \\ \mathsf{h}(\mathsf{1}(\mathsf{0}(x, y), z)) &\to \mathsf{0}(x, \mathsf{1}(z, \mathsf{h}(y))) \end{aligned}$$

Not exhaustive, since there is no rule for $\mathsf{h}(\mathsf{1}(\mathsf{1}(\ldots, \ldots), \ldots))$.

Productivity

In the sequel, let R be an orthogonal and exhaustive constructor TRS.

Definition

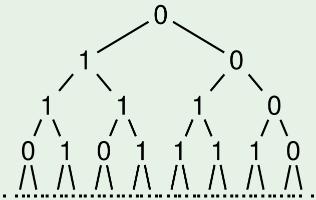
R is productive for a term t if $t \twoheadrightarrow\!\!\!\!\rightarrow s$ with $s \in Ter^{\infty}(\mathcal{C}(R), \varnothing)$.

Proposition

R is productive for a term t if and only if for every $n \in \mathbb{N}$ there exists a rewrite sequence $t \twoheadrightarrow s$ with s consisting up to depth n only of $\mathcal{C}(R)$.

Example

$$\begin{aligned} \mathsf{T} &\to \mathsf{0}(\mathsf{h}(\mathsf{T},\mathsf{T}),\mathsf{T}) \\ \mathsf{h}(\mathsf{0}(x,y),z) &\to \mathsf{1}(\mathsf{h}(z,x),\mathsf{h}(y,x)) \\ \mathsf{h}(\mathsf{1}(x,y),z) &\to \mathsf{0}(\mathsf{h}(z,x),\mathsf{h}(y,x)) \end{aligned}$$



Global Productivity

Definition (Global Productivity)

R is globally productive if R is productive for every $t \in Ter(\Sigma, \varnothing)$.

Example

$$
\begin{aligned}
\mathsf{Alt} &\to 0 : 1 : \mathsf{Alt} \\
\mathsf{Zeros} &\to \mathsf{filter0}(\mathsf{Alt}) \\
\mathsf{filter0}(0 : x) &\to 0 : \mathsf{filter0}(x) \\
\mathsf{filter0}(1 : x) &\to \mathsf{filter0}(x)
\end{aligned}
\qquad
\begin{aligned}
\mathsf{Ones} &\to \mathsf{filter1}(\mathsf{Alt}) \\
\mathsf{filter1}(0 : x) &\to \mathsf{filter1}(x) \\
\mathsf{filter1}(1 : x) &\to 1 : \mathsf{filter1}(x)
\end{aligned}
$$

Not globally productive since filter0(Ones) is not productive.

Proposition ([ZR10])

R is globally productive if and only if every $t \in Ter(\Sigma, \varnothing)$ admits a rewrite sequence $t \twoheadrightarrow \mathsf{c}(t_1, \ldots, t_n)$ with $c \in \mathcal{C}(R)$.

Friendly Nesting

Definition (Shallow TRS)

R is called shallow if for every rule $\mathsf{f}(t_1, \ldots, t_n) \to r$ and $i = 1, \ldots, n$:

- $t_i \in \mathcal{X}$, or
- $t_i = \mathsf{c}(x_1, \ldots, x_m)$ with $x_1, \ldots, x_m \in \mathcal{X}$.

(that is, the maximal consumption for every argument is 1)

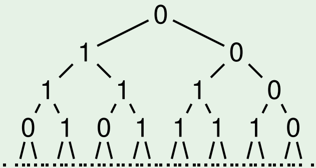
A simple criterion for productivity:

Theorem (Friendly Nesting [EGH08, ZR10])

A shallow R is globally productive if $root(r) \in \mathcal{C}(R)$ for all $\ell \to r \in R$.

Example (A Friendly Nesting Specification)

$$
\begin{aligned}
\mathsf{T} &\to \mathsf{0}(\mathsf{h}(\mathsf{T},\mathsf{T}),\mathsf{T}) \\
\mathsf{h}(\mathsf{0}(x,y),z) &\to \mathsf{1}(\mathsf{h}(z,x),\mathsf{h}(y,x)) \\
\mathsf{h}(\mathsf{1}(x,y),z) &\to \mathsf{0}(\mathsf{h}(z,x),\mathsf{h}(y,x))
\end{aligned}
$$



Non-Productivity Preserving Transformations

Theorem ([ZR10])

Let R' be obtained from R by:

- *replacing $\ell \to r \in R$ by $\ell \to r'$ such that $r \to_R^* r'$.*

Then R is productive if and only if R' is productive.

Example

$$
\begin{aligned}
\mathsf{T} &\to \mathsf{h}(\mathsf{0}(\mathsf{T},\mathsf{T}),\mathsf{T}) \\
\mathsf{h}(\mathsf{0}(x,y),z) &\to \mathsf{1}(\mathsf{h}(z,x),\mathsf{h}(y,x)) \\
\mathsf{h}(\mathsf{1}(x,y),z) &\to \mathsf{0}(\mathsf{h}(z,x),\mathsf{h}(y,x))
\end{aligned}
$$

We can replace the first rule by:

$$\mathsf{T} \to \mathsf{1}(\mathsf{h}(\mathsf{T},\mathsf{T}),\mathsf{h}(\mathsf{T},\mathsf{T}))$$

Then the specification is friendly nesting, and hence productive.
As a consequence also the original specification is productive.

Proving data-aware productivity via **context-sensitive termination**.

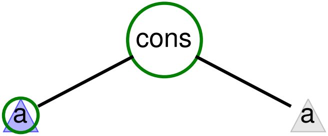
What is Context-Sensitive Term Rewriting?

- replacement map $\mu(f) \subseteq \{1, \ldots, ar(f)\}$ defines in which arguments may be reduced.

Example

$$a \rightarrow \text{cons}(0, a)$$

with $\mu(cons) = \{1\}$.



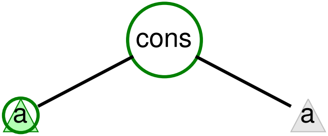
What is Context-Sensitive Term Rewriting?

- replacement map $\mu(f) \subseteq \{1, \ldots, ar(f)\}$ defines in which arguments may be reduced.

Example

$$a \to \mathrm{cons}(0, a)$$

with $\mu(cons) = \{1\}$.



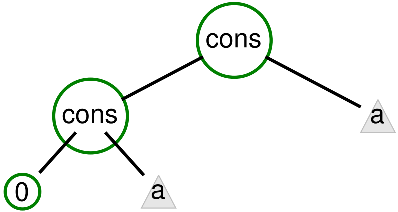
What is Context-Sensitive Term Rewriting?

- replacement map $\mu(f) \subseteq \{1, \dots, ar(f)\}$ defines in which arguments may be reduced.

Example

$$a \to \text{cons}(0, a)$$

with $\mu(cons) = \{1\}$.

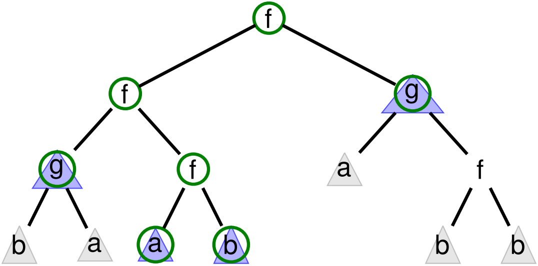


What is Context-Sensitive Term Rewriting?

Example

$$a \rightarrow a \qquad b \rightarrow b \qquad \mathrm{g}(x, y) \rightarrow \mathrm{f}(x, y)$$

with $\mu(f) = \{1, 2\}$, $\mu(g) = \{\}$.



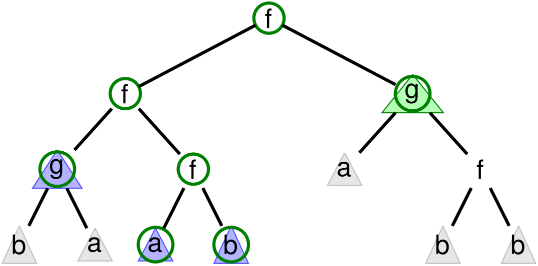
By $\mu(g) = \{\}$ we forbid rewriting below g.

What is Context-Sensitive Term Rewriting?

Example

$$a \to a \qquad b \to b \qquad g(x, y) \to f(x, y)$$

with $\mu(f) = \{1, 2\}$, $\mu(g) = \{\}$.



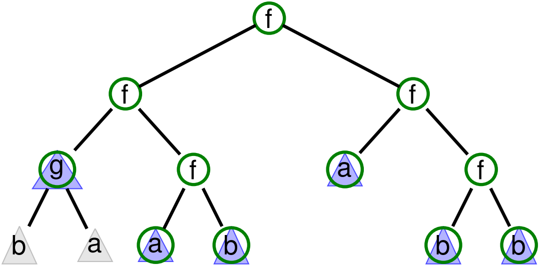
By $\mu(g) = \{\}$ we forbid rewriting below g.

What is Context-Sensitive Term Rewriting?

Example

$$a \to a \qquad b \to b \qquad \mathrm{g}(x, y) \to \mathrm{f}(x, y)$$

with $\mu(f) = \{1, 2\}$, $\mu(g) = \{\}$.



By $\mu(g) = \{\}$ we forbid rewriting below g.

Productivity via Context-Sensitive Termination

Theorem ([ZR10])

Let R be shallow. Let R_μ be defined as R with additionally:

$$\begin{aligned} \mu(\mathsf{f}) &= \{1, \ldots, \mathsf{ar}(\mathsf{f})\} && \text{for every } \mathsf{f} \in \mathcal{D}(R) \\ \mu(\mathsf{c}) &= \varnothing && \text{for every } \mathsf{c} \in \mathcal{C}(R) \end{aligned}$$

Then R is productive if R_μ is context-sensitive terminating.

Productivity via Context-Sensitive Termination

Theorem ([ZR10])

Let R be shallow. Let R_μ be defined as R with additionally:

$$\mu(\mathsf{f}) = \{1, \ldots, \mathsf{ar}(\mathsf{f})\} \qquad \text{for every } \mathsf{f} \in \mathcal{D}(R)$$

$$\mu(\mathsf{c}) = \varnothing \qquad \text{for every } \mathsf{c} \in \mathcal{C}(R)$$

Then R is productive if R_μ is context-sensitive terminating.

The method allows for proving data-aware productivity:

Example

$$\mathsf{T} \to \mathsf{f}(\mathsf{0}(\mathsf{T})) \qquad \mathsf{f}(\mathsf{0}(x)) \to \mathsf{0}(\mathsf{f}(x))$$

$$\mathsf{f}(\mathsf{1}(x)) \to \mathsf{f}(x)$$

Productivity via Context-Sensitive Termination

Theorem ([ZR10])

Let R be shallow. Let R_μ be defined as R with additionally:

$$\mu(\mathsf{f}) = \{1, \ldots, \mathsf{ar}(\mathsf{f})\} \qquad \textit{for every } \mathsf{f} \in \mathcal{D}(R)$$
$$\mu(\mathsf{c}) = \varnothing \qquad \textit{for every } \mathsf{c} \in \mathcal{C}(R)$$

Then R is productive if R_μ is context-sensitive terminating.

The method allows for proving data-aware productivity:

Example

$$\mathsf{T} \to \mathsf{f}(\mathsf{0}(\mathsf{T})) \qquad \mathsf{f}(\mathsf{0}(x)) \to \mathsf{0}(\mathsf{f}(x))$$
$$\mathsf{f}(\mathsf{1}(x)) \to \mathsf{f}(x)$$

We use the following replacement map:

$$\mu(\mathsf{f}) = \{1\} \qquad \mu(\mathsf{0}) = \varnothing \qquad \mu(\mathsf{1}) = \varnothing$$

Productivity via Context-Sensitive Termination

Theorem ([ZR10])

Let R be shallow. Let R_μ be defined as R with additionally:

$$\begin{aligned} \mu(\mathsf{f}) &= \{1, \ldots, \mathsf{ar}(\mathsf{f})\} && \textit{for every } \mathsf{f} \in \mathcal{D}(R) \\ \mu(\mathsf{c}) &= \varnothing && \textit{for every } \mathsf{c} \in \mathcal{C}(R) \end{aligned}$$

Then R is productive if R_μ is context-sensitive terminating.

The method allows for proving data-aware productivity:

Example

$$\mathsf{T} \to \mathsf{f}(\mathsf{0}(\mathsf{T})) \qquad \begin{aligned} \mathsf{f}(\mathsf{0}(x)) &\to \mathsf{0}(\mathsf{f}(x)) \\ \mathsf{f}(\mathsf{1}(x)) &\to \mathsf{f}(x) \end{aligned}$$

We use the following replacement map:

$$\mu(\mathsf{f}) = \{1\} \qquad \mu(\mathsf{0}) = \varnothing \qquad \mu(\mathsf{1}) = \varnothing$$

The obtained system is context-sensitive terminating.

Productivity via Context-Sensitive Termination

Theorem ([ZR10])

Let R be shallow. Let R_μ be defined as R with additionally:

$$\mu(\mathsf{f}) = \{1, \ldots, \mathsf{ar}(\mathsf{f})\} \qquad \textit{for every } \mathsf{f} \in \mathcal{D}(R)$$
$$\mu(\mathsf{c}) = \varnothing \qquad \textit{for every } \mathsf{c} \in \mathcal{C}(R)$$

Then R is productive if R_μ is context-sensitive terminating.

The method allows for proving data-aware productivity:

Example

$$\mathsf{T} \to \mathsf{f}(\mathsf{0}(\mathsf{T})) \qquad \mathsf{f}(\mathsf{0}(x)) \to \mathsf{0}(\mathsf{f}(x))$$
$$\mathsf{f}(\mathsf{1}(x)) \to \mathsf{f}(x)$$

We use the following replacement map:

$$\mu(\mathsf{f}) = \{1\} \qquad \mu(\mathsf{0}) = \varnothing \qquad \mu(\mathsf{1}) = \varnothing$$

The obtained system is context-sensitive terminating.
Hence the specification is globally productive

Productivity via Context-Sensitive Termination

Limitations of the method:

Example

$$\begin{aligned} \mathsf{W} &\to \mathsf{zip}(\mathsf{0}(\mathsf{W}), \mathsf{W}) \\ \mathsf{zip}(\mathsf{0}(\sigma), \tau) &\to \mathsf{0}(\mathsf{zip}(\tau, \sigma)) \\ \mathsf{zip}(\mathsf{1}(\sigma), \tau) &\to \mathsf{1}(\mathsf{zip}(\tau, \sigma)) \end{aligned}$$

Productivity via Context-Sensitive Termination

Limitations of the method:

Example

$$\mathsf{W} \to \mathsf{zip}(\mathsf{0}(\mathsf{W}), \mathsf{W})$$
$$\mathsf{zip}(\mathsf{0}(\sigma), \tau) \to \mathsf{0}(\mathsf{zip}(\tau, \sigma))$$
$$\mathsf{zip}(\mathsf{1}(\sigma), \tau) \to \mathsf{1}(\mathsf{zip}(\tau, \sigma))$$

We use the following replacement map:

$$\mu(\mathsf{zip}) = \{1, 2\} \qquad \mu(\mathsf{0}) = \varnothing \qquad \mu(\mathsf{1}) = \varnothing$$

Productivity via Context-Sensitive Termination

Limitations of the method:

Example

$$\begin{aligned} \mathsf{W} &\to \mathsf{zip}(\mathsf{0}(\mathsf{W}), \mathsf{W}) \\ \mathsf{zip}(\mathsf{0}(\sigma), \tau) &\to \mathsf{0}(\mathsf{zip}(\tau, \sigma)) \\ \mathsf{zip}(\mathsf{1}(\sigma), \tau) &\to \mathsf{1}(\mathsf{zip}(\tau, \sigma)) \end{aligned}$$

We use the following replacement map:

$$\mu(\mathsf{zip}) = \{1, 2\} \qquad \mu(\mathsf{0}) = \varnothing \qquad \mu(\mathsf{1}) = \varnothing$$

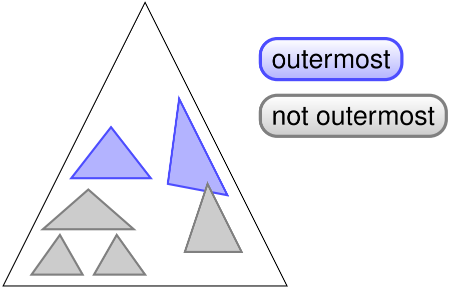
The obtained system is not context-sentitive terminating:

$$\mathsf{W} \to \mathsf{zip}(\mathsf{0}(\mathsf{W}), \mathsf{W}) \to \mathsf{zip}(\mathsf{0}(\mathsf{W}), \mathsf{zip}(\mathsf{0}(\mathsf{W}), \mathsf{W})) \to \ldots$$

Proving data-aware productivity via **outermost termination**.

What is Outermost Term Rewriting?

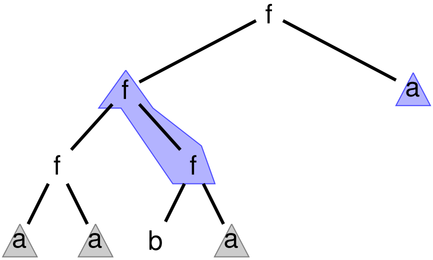
- Only outermost redexes may be reduced.
- Outermost = not below another redex position.



What is Outermost Term Rewriting?

Example

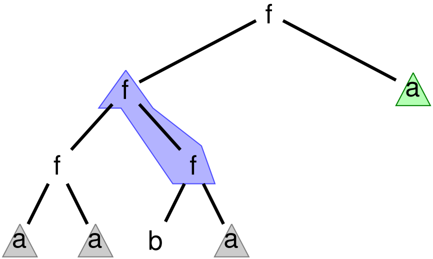
$a \to \mathsf{f}(b, a)$ $\qquad$ $\mathsf{f}(x, \mathsf{f}(y, z)) \to b$



What is Outermost Term Rewriting?

Example

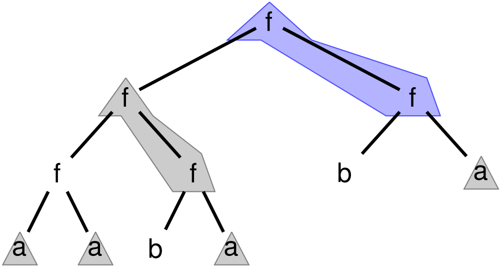
$a \rightarrow \mathsf{f}(b, a) \qquad \mathsf{f}(x, \mathsf{f}(y, z)) \rightarrow b$



What is Outermost Term Rewriting?

Example

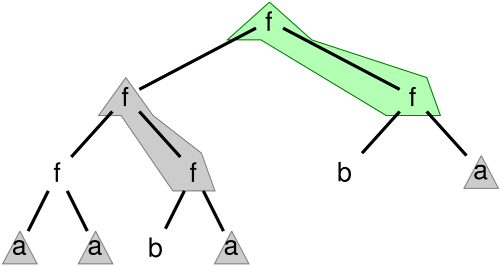
$$a \rightarrow \mathsf{f}(b, a) \qquad \mathsf{f}(x, \mathsf{f}(y, z)) \rightarrow b$$



What is Outermost Term Rewriting?

Example

$a \to \mathsf{f}(b, a)$ $\quad\quad$ $\mathsf{f}(x, \mathsf{f}(y, z)) \to b$



What is Outermost Term Rewriting?

Example

$$a \to \mathsf{f}(b, a) \qquad\qquad \mathsf{f}(x, \mathsf{f}(y, z)) \to b$$

b

Not terminating, but outermost terminating.

What is Outermost Term Rewriting?

Example

$$
\begin{aligned}
a &\to \mathsf{f}(a, a) & \qquad \mathsf{f}(\mathsf{f}(x, y), \mathsf{f}(z, w)) &\to a \\
\mathsf{f}(\mathsf{f}(x, y), x) &\to b & \qquad \mathsf{f}(x, \mathsf{f}(y, z)) &\to b
\end{aligned}
$$

What is Outermost Term Rewriting?

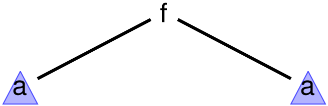
Example

$$
\begin{aligned}
a &\to \mathsf{f}(a, a) & \mathsf{f}(\mathsf{f}(x, y), \mathsf{f}(z, w)) &\to a \\
\mathsf{f}(\mathsf{f}(x, y), x) &\to b & \mathsf{f}(x, \mathsf{f}(y, z)) &\to b
\end{aligned}
$$

What is Outermost Term Rewriting?

Example

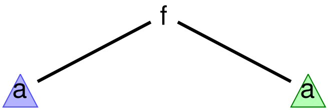
$$
\begin{aligned}
a &\to \mathsf{f}(a, a) & \mathsf{f}(\mathsf{f}(x, y), \mathsf{f}(z, w)) &\to a \\
\mathsf{f}(\mathsf{f}(x, y), x) &\to b & \mathsf{f}(x, \mathsf{f}(y, z)) &\to b
\end{aligned}
$$



What is Outermost Term Rewriting?

Example

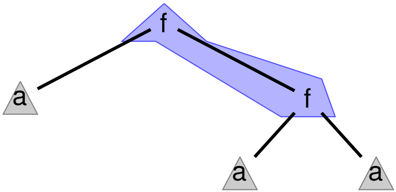
$$a \to \mathsf{f}(a, a) \qquad \mathsf{f}(\mathsf{f}(x, y), \mathsf{f}(z, w)) \to a$$
$$\mathsf{f}(\mathsf{f}(x, y), x) \to b \qquad \mathsf{f}(x, \mathsf{f}(y, z)) \to b$$



What is Outermost Term Rewriting?

Example

$$a \to f(a, a) \qquad f(f(x, y), f(z, w)) \to a$$
$$f(f(x, y), x) \to b \qquad f(x, f(y, z)) \to b$$

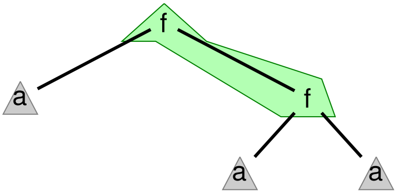


What is Outermost Term Rewriting?

Example

$$a \to \mathsf{f}(a, a) \qquad \mathsf{f}(\mathsf{f}(x, y), \mathsf{f}(z, w)) \to a$$

$$\mathsf{f}(\mathsf{f}(x, y), x) \to b \qquad \mathsf{f}(x, \mathsf{f}(y, z)) \to b$$



What is Outermost Term Rewriting?

Example

$$a \to \mathsf{f}(a, a) \qquad \mathsf{f}(\mathsf{f}(x, y), \mathsf{f}(z, w)) \to a$$
$$\mathsf{f}(\mathsf{f}(x, y), x) \to b \qquad \mathsf{f}(x, \mathsf{f}(y, z)) \to b$$

b

Not terminating, but outermost terminating.

Productivity via Outermost Termination

Theorem (Generalisation of [ZR09])

Let R be shallow. Let $R_\perp$ be the extension of R with rules:

$$\mathsf{c}(x_1, \ldots, x_n) \to \perp \qquad\qquad \textit{for every } \mathsf{c} \in \mathcal{C}(R)$$

Then R is productive if $R_\perp$ is outermost terminating.

Productivity via Outermost Termination

Theorem (Generalisation of [ZR09])

Let R be shallow. Let $R_\perp$ be the extension of R with rules:

$$\mathsf{c}(x_1, \ldots, x_n) \to \perp \qquad \textit{for every } \mathsf{c} \in \mathcal{C}(R)$$

Then R is productive if $R_\perp$ is outermost terminating.

The method allows for proving data-aware productivity:

Example

$$\mathsf{T} \to \mathsf{f}(\mathsf{0}(\mathsf{T})) \qquad \mathsf{f}(\mathsf{0}(x)) \to \mathsf{0}(\mathsf{f}(x))$$

$$\mathsf{f}(\mathsf{1}(x)) \to \mathsf{f}(x)$$

Productivity via Outermost Termination

Theorem (Generalisation of [ZR09])

Let R be shallow. Let $R_\perp$ be the extension of R with rules:

$$\mathsf{c}(x_1, \ldots, x_n) \to \perp \qquad \textit{for every } \mathsf{c} \in \mathcal{C}(R)$$

Then R is productive if $R_\perp$ is outermost terminating.

The method allows for proving data-aware productivity:

Example

$$\mathsf{T} \to \mathsf{f}(\mathsf{0}(\mathsf{T})) \qquad \mathsf{f}(\mathsf{0}(x)) \to \mathsf{0}(\mathsf{f}(x))$$
$$\mathsf{f}(\mathsf{1}(x)) \to \mathsf{f}(x)$$

We add overflow rules:

$$\mathsf{0}(\sigma) \to \perp \qquad \mathsf{1}(\sigma) \to \perp$$

Productivity via Outermost Termination

Theorem (Generalisation of [ZR09])

Let R be shallow. Let $R_\perp$ be the extension of R with rules:

$$\mathsf{c}(x_1, \ldots, x_n) \to \bot \qquad \textit{for every } \mathsf{c} \in \mathcal{C}(R)$$

Then R is productive if $R_\perp$ is outermost terminating.

The method allows for proving data-aware productivity:

Example

$$\mathsf{T} \to \mathsf{f}(0(\mathsf{T})) \qquad \mathsf{f}(0(x)) \to 0(\mathsf{f}(x))$$
$$\mathsf{f}(1(x)) \to \mathsf{f}(x)$$

We add overflow rules:

$$0(\sigma) \to \bot \qquad 1(\sigma) \to \bot$$

The obtained system is outermost terminating.

Productivity via Outermost Termination

Theorem (Generalisation of [ZR09])

Let R be shallow. Let $R_\perp$ be the extension of R with rules:

$$\mathsf{c}(x_1, \ldots, x_n) \to \perp \qquad \textit{for every } \mathsf{c} \in \mathcal{C}(R)$$

Then R is productive if $R_\perp$ is outermost terminating.

The method allows for proving data-aware productivity:

Example

$$\mathsf{T} \to \mathsf{f}(\mathsf{0}(\mathsf{T})) \qquad \begin{aligned} \mathsf{f}(\mathsf{0}(x)) &\to \mathsf{0}(\mathsf{f}(x)) \\ \mathsf{f}(\mathsf{1}(x)) &\to \mathsf{f}(x) \end{aligned}$$

We add overflow rules:

$$\mathsf{0}(\sigma) \to \perp \qquad \mathsf{1}(\sigma) \to \perp$$

The obtained system is outermost terminating.
Hence the specification is globally productive

Productivity via Outermost Termination

Example

$$\begin{aligned} \mathsf{W} &\to \mathsf{zip}(\mathsf{0}(\mathsf{W}), \mathsf{W}) \\ \mathsf{zip}(\mathsf{0}(\sigma), \tau) &\to \mathsf{0}(\mathsf{zip}(\tau, \sigma)) \\ \mathsf{zip}(\mathsf{1}(\sigma), \tau) &\to \mathsf{1}(\mathsf{zip}(\tau, \sigma)) \end{aligned}$$

Productivity via Outermost Termination

Example

$$\begin{aligned} \mathsf{W} &\to \mathsf{zip}(\mathsf{0}(\mathsf{W}), \mathsf{W}) \\ \mathsf{zip}(\mathsf{0}(\sigma), \tau) &\to \mathsf{0}(\mathsf{zip}(\tau, \sigma)) \\ \mathsf{zip}(\mathsf{1}(\sigma), \tau) &\to \mathsf{1}(\mathsf{zip}(\tau, \sigma)) \end{aligned}$$

We add overflow rules:

$$\mathsf{0}(\sigma) \to \bot \qquad\qquad \mathsf{1}(\sigma) \to \bot$$

Productivity via Outermost Termination

Example

$$\begin{aligned} \mathsf{W} &\to \mathsf{zip}(\mathsf{0}(\mathsf{W}), \mathsf{W}) \\ \mathsf{zip}(\mathsf{0}(\sigma), \tau) &\to \mathsf{0}(\mathsf{zip}(\tau, \sigma)) \\ \mathsf{zip}(\mathsf{1}(\sigma), \tau) &\to \mathsf{1}(\mathsf{zip}(\tau, \sigma)) \end{aligned}$$

We add overflow rules:

$$\mathsf{0}(\sigma) \to \bot \qquad\qquad \mathsf{1}(\sigma) \to \bot$$

The obtained system is outermost terminating.

Productivity via Outermost Termination

Example

$$\mathsf{W} \to \mathsf{zip}(0(\mathsf{W}), \mathsf{W})$$
$$\mathsf{zip}(0(\sigma), \tau) \to 0(\mathsf{zip}(\tau, \sigma))$$
$$\mathsf{zip}(1(\sigma), \tau) \to 1(\mathsf{zip}(\tau, \sigma))$$

We add overflow rules:

$$0(\sigma) \to \bot \qquad\qquad 1(\sigma) \to \bot$$

The obtained system is outermost terminating.
Hence the specification is globally productive

Productivity via Outermost Termination

Limitations of the method:

Example

$$\begin{aligned} \mathsf{D} &\to \mathsf{zip}(\mathsf{Alt}, \mathsf{D}) \\ \mathsf{Alt} &\to 0(1(\mathsf{Alt})) \\ \mathsf{zip}(0(\sigma), \tau) &\to 0(\mathsf{zip}(\tau, \sigma)) \\ \mathsf{zip}(1(\sigma), \tau) &\to 1(\mathsf{zip}(\tau, \sigma)) \end{aligned}$$

Productivity via Outermost Termination

Limitations of the method:

Example

$$\begin{aligned} \mathsf{D} &\to \mathsf{zip}(\mathsf{Alt}, \mathsf{D}) \\ \mathsf{Alt} &\to \mathsf{0}(\mathsf{1}(\mathsf{Alt})) \\ \mathsf{zip}(\mathsf{0}(\sigma), \tau) &\to \mathsf{0}(\mathsf{zip}(\tau, \sigma)) \\ \mathsf{zip}(\mathsf{1}(\sigma), \tau) &\to \mathsf{1}(\mathsf{zip}(\tau, \sigma)) \end{aligned}$$

We add overflow rules:

$$\mathsf{0}(\sigma) \to \bot \qquad\qquad \mathsf{1}(\sigma) \to \bot$$

Productivity via Outermost Termination

Limitations of the method:

Example

$$\begin{aligned} \mathsf{D} &\to \mathsf{zip}(\mathsf{Alt}, \mathsf{D}) \\ \mathsf{Alt} &\to \mathsf{0}(\mathsf{1}(\mathsf{Alt})) \\ \mathsf{zip}(\mathsf{0}(\sigma), \tau) &\to \mathsf{0}(\mathsf{zip}(\tau, \sigma)) \\ \mathsf{zip}(\mathsf{1}(\sigma), \tau) &\to \mathsf{1}(\mathsf{zip}(\tau, \sigma)) \end{aligned}$$

We add overflow rules:

$$\mathsf{0}(\sigma) \to \bot \qquad\qquad \mathsf{1}(\sigma) \to \bot$$

The obtained system is not outermost terminating:

$$\mathsf{D} \to \mathsf{zip}(\mathsf{Alt}, \mathsf{D}) \to \mathsf{zip}(\mathsf{Alt}, \mathsf{zip}(\mathsf{Alt}, \mathsf{D})) \to \dots$$

Summary

We have seen:

- friendly nesting: a simple criterion for productivity
- rewriting right-hand sides
- transformations from productivity to termination:
  - to context-sensitive, and
  - to outermost rewriting.

These latter methods allow for proving data-aware productivity!

Bibliography

[EGH08] Endrullis, Grabmayer and Hendriks
Data-Oblivious Stream Productivity, LPAR 2008

[EH09] Endrullis and Hendriks
From Outermost to Context-Sensitive Rewriting, RTA 2009

[ZR08] Zantema and Raffelsieper
A transformational approach to prove outermost termination automatically, WRS 2008

[ZR09] Zantema and Raffelsieper
Stream Productivity by Outermost Termination, WRS 2009

[ZR10] Zantema and Raffelsieper
Proving Productivity in Infinite Data Structures, RTA 2010