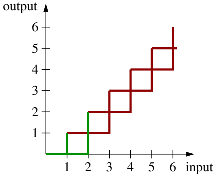
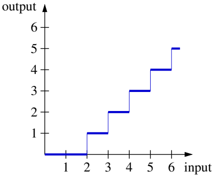
Part 4: Data-Oblivious Productivity

Jörg Endrullis Clemens Grabmayer Dimitri Hendriks

Vrije Universiteit Amsterdam – Universiteit Utrecht – Vrije Universiteit Amsterdam





ISR 2010, Utrecht University

July 8, 2010

Course Overview

Yesterday:

1. introduction, history (D)
2. the pure stream format, pebbleflow nets, decidability (D)
3. extended formats (C)

Today:

4. data-oblivious productivity (C)
5. productivity of infinite data structures via termination (J)
6. complexity and variants of productivity (C)
7. practicum: defining streams (you)

Course Overview

Yesterday:

1. introduction, history (D)
2. the pure stream format, pebbleflow nets, decidability (D)
3. extended formats (C)

Today:

4. data-oblivious productivity (C)
5. productivity of infinite data structures via termination (J)
6. complexity and variants of productivity (C)
7. practicum: defining streams (you)

Course Overview

Yesterday:

1. introduction, history (D)
2. the pure stream format, pebbleflow nets, decidability (D)
3. extended formats (C)

Today:

4. data-oblivious productivity (C)
5. productivity of infinite data structures via termination (J)
6. complexity and variants of productivity (C)
7. practicum: defining streams (you)

Course Overview

Yesterday:

1. introduction, history (D)
2. the pure stream format, pebbleflow nets, decidability (D)
3. extended formats (C)

Today:

4. data-oblivious productivity (C)
5. productivity of infinite data structures via termination (J)
6. complexity and variants of productivity (C)
7. practicum: defining streams (you)

Course Overview

Yesterday:

1. introduction, history (D)
2. the pure stream format, pebbleflow nets, decidability (D)
3. extended formats (C)

Today:

4. data-oblivious productivity (C)
5. productivity of infinite data structures via termination (J)
6. complexity and variants of productivity (C)
7. practicum: defining streams (you)

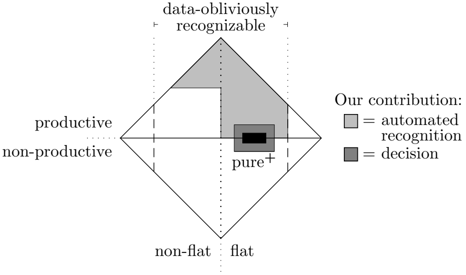
Overview

1. Data-Oblivious Analysis

2. Gathered Results for PSF-Extensions

PSF-extensions, and preview of results

- for pure$^+$ stream spec's: a decision method for productivity
- for flat stream spec's: a computable criterion for productivity that is "data-obliviously optimal"
- for friendly nesting stream spec's: a computable criterion for productivity



Overview

1. Data-Oblivious Analysis

2. Gathered Results for PSF-Extensions

Data-Oblivious Analysis

Example (Pascal's triangle)

$$\mathsf{P} \to 0 : \mathsf{s}(0) : \mathsf{g}(\mathsf{P})$$
$$\mathsf{g}(\mathsf{s}(x) : y : xs) \to \mathsf{a}(\mathsf{s}(x), y) : \mathsf{g}(y : xs)$$
$$\mathsf{g}(0 : xs) \to 0 : \mathsf{s}(0) : \mathsf{g}(xs)$$

By data abstraction:

$$\mathsf{P}' \to \bullet : \bullet : \mathsf{g}(\mathsf{P}')$$
$$\mathsf{g}(\bullet : \bullet : xs) \to \bullet : \mathsf{g}(\bullet : xs)$$
$$\mathsf{g}(\bullet : xs) \to \bullet : \bullet : \mathsf{g}(xs)$$

The data oblivious lower/upper bounds on the production of g are:

$$n \mapsto n \dot{-} 1 \;/\; n \mapsto 2n$$

The lower bound implies productivity of P'. One can say:
P is data-obliviously productive. This clearly implies productivity of P.

Data-Oblivious Rewriting

formalised by a two-player game between:

- a data-exchange player D can exchange data elements arbitrarily
- a rewrite player R can perform usual term rewriting steps

player D can help or handicap the rewrite player

⇒ for a d-o analysis we have to quantify over all strategies of D

Definition (data-oblivious lower bound on the production of a term s)

$$\underline{do}_{\mathcal{R}}(s) := \text{worst case production of } s \text{ (number of elements) by quantification over all strategies for D}$$

A stream spec $\mathcal{R}$ is data-obliviously productive if $\underline{do}_{\mathcal{R}}(\mathsf{M}_0) = \infty$.

Data-Oblivious Rewriting

formalised by a two-player game between:

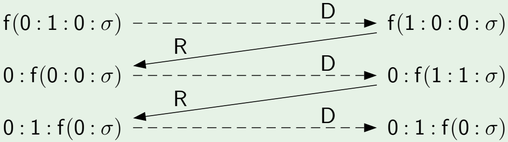
- a data-exchange player D can exchange data elements arbitrarily
- a rewrite player R can perform usual term rewriting steps

player D can help or handicap the rewrite player

$\Rightarrow$ for a d-o analysis we have to quantify over all strategies of D

Example ($\mathsf{f}(0 : x : xs) \to x : x : \mathsf{f}(0 : xs),\ \mathsf{f}(1 : x : xs) \to x : \mathsf{f}(0 : xs)$)

Data-oblivious rewriting of the term $\mathsf{f}(0 : 1 : 0 : xs)$:



Data-Oblivious Rewriting

formalised by a two-player game between:

- a data-exchange player D can exchange data elements arbitrarily
- a rewrite player R can perform usual term rewriting steps

player D can help or handicap the rewrite player

$\Rightarrow$ for a d-o analysis we have to quantify over all strategies of D

Definition (data-oblivious lower bound on the production of a term *s*)

$\underline{do}_{\mathcal{R}}(s)$:= worst case production of *s* (number of elements) by quantification over all strategies for D

A stream spec $\mathcal{R}$ is data-obliviously productive if $\underline{do}_{\mathcal{R}}(\mathsf{M}_0) = \infty$.

Data-Oblivious Productivity

Definition

The data-oblivious production range ($\subseteq \overline{\mathbb{N}}$) of a term t:

$$\underline{\overline{do}}_{\mathcal{R}}(t) \;:=\; \text{set of all productions of } t \text{ under outermost-fair data-oblivious rewrite sequences starting at } t$$

The d-o lower/upper bounds:

$$\underline{do}_{\mathcal{R}}(t) := \inf \underline{\overline{do}}_{\mathcal{R}}(t) \qquad \overline{do}_{\mathcal{R}}(t) := \sup \underline{\overline{do}}_{\mathcal{R}}(t)$$

A term t is data-obliviously productive if $\underline{do}_{\mathcal{R}}(t) = \infty$.

$\Pi_{\mathcal{R}}(t) := \sup\{n \in \overline{\mathbb{N}} \mid t \twoheadrightarrow s_1 : \ldots : s_n : r\}$ data-aware production of t.

Data-Oblivious Productivity

Definition

The data-oblivious production range ($\subseteq \overline{\mathbb{N}}$) of a term t:

$\overline{\underline{do}}_{\mathcal{R}}(t)$:= set of all productions of t under outermost-fair data-oblivious rewrite sequences starting at t

The d-o lower/upper bounds:

$$\underline{do}_{\mathcal{R}}(t) := \inf \overline{\underline{do}}_{\mathcal{R}}(t) \qquad \overline{do}_{\mathcal{R}}(t) := \sup \overline{\underline{do}}_{\mathcal{R}}(t)$$

A term t is data-obliviously productive if $\underline{do}_{\mathcal{R}}(t) = \infty$.

$\Pi_{\mathcal{R}}(t) := \sup\{n \in \overline{\mathbb{N}} \mid t \twoheadrightarrow s_1 : \ldots : s_n : r\}$ data-aware production of t.

Data-Oblivious Productivity versus Productivity

Proposition

Let $\mathcal{R} = \langle \Sigma, R \rangle$ be a stream specification.

- *For all stream terms $s \in Ter(\Sigma)_S$:*

$$\underline{do}_{\mathcal{R}}(s) \leq \Pi_{\mathcal{R}}(s) \leq \overline{do}_{\mathcal{R}}(s) \; .$$

A stream specification $\mathcal{R}$ is called:

- data-obliviously productive if $\underline{do}_{\mathcal{R}}(\mathsf{M}_0) = \infty$;
- data-obliviously non-productive if $\overline{do}_{\mathcal{R}}(\mathsf{M}_0) < \infty$.

Theorem

- *$\mathcal{R}$ data-obliviously productive $\implies$ $\mathcal{R}$ productive;*
- *$\mathcal{R}$ data-obliviously non-productive $\implies$ $\mathcal{R}$ not productive;*

Data-Oblivious Productivity versus Productivity

Proposition

Let $\mathcal{R} = \langle \Sigma, R \rangle$ be a stream specification.

- *For all stream terms $s \in Ter(\Sigma)_S$:*

$$\underline{do}_{\mathcal{R}}(s) \leq \Pi_{\mathcal{R}}(s) \leq \overline{do}_{\mathcal{R}}(s) \ .$$

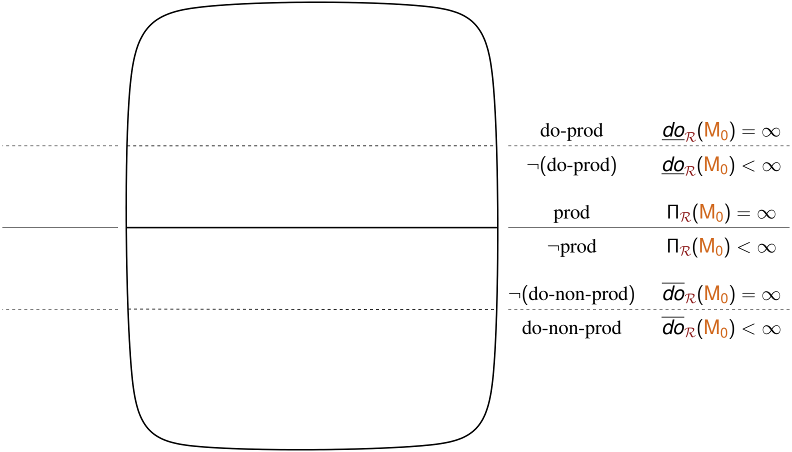
A stream specification $\mathcal{R}$ is called:

- data-obliviously productive if $\underline{do}_{\mathcal{R}}(\mathsf{M}_0) = \infty$;
- data-obliviously non-productive if $\overline{do}_{\mathcal{R}}(\mathsf{M}_0) < \infty$.

Theorem

- *$\mathcal{R}$ data-obliviously productive $\implies$ $\mathcal{R}$ productive;*
- *$\mathcal{R}$ data-obliviously non-productive $\implies$ $\mathcal{R}$ not productive;*

Data-Oblivious Productivity versus Productivity

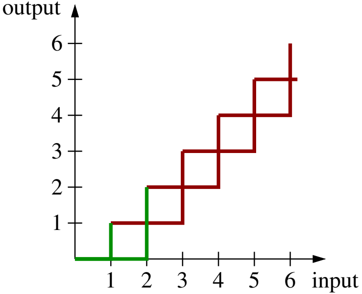


D-O Lower Bounds of Stream Functions

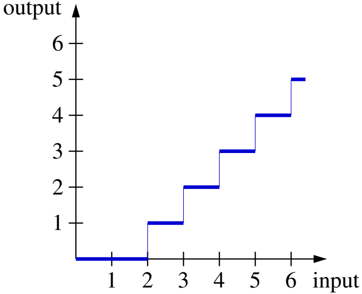
In this approach exact data-oblivious lower bounds are computed:

$$\mathbf{f}(0 : xs) \to 0 : \mathsf{read2}(xs) \qquad \mathbf{f}(1 : x : xs) \to 1 : x : \mathsf{read2}(xs)$$

$$\mathsf{read2}(x : y : xs) \to x : y : \mathsf{read2}(xs)$$



2 possible function-call traces for **f**



exact lower bound $\underline{do}_{\mathcal{R}}(\mathbf{f})$

Example, Limits of Data-Oblivious Analysis

Example

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

This specification is productive:

$$\mathsf{T} \twoheadrightarrow 1 : \mathsf{f}(\mathsf{T}) \twoheadrightarrow 1 : 1 : \mathsf{f}(\mathsf{f}(\mathsf{T})) \twoheadrightarrow \ldots \twoheadrightarrow\!\!\!\!\!\!\twoheadrightarrow 1 : 1 : 1 : 1 : \ldots \;,$$

but, disregarding the identity of data, the rewrite sequence:

$$\mathsf{T} \to \mathsf{f}(\bullet : \mathsf{T}) \to \mathsf{f}(\mathsf{T}) \twoheadrightarrow \ldots \twoheadrightarrow\!\!\!\!\!\!\twoheadrightarrow \mathsf{f}(\mathsf{f}(\mathsf{f}(\ldots))) \;.$$

is possible. Hence the specification is not data-obliviously productive (i.e., productivity of this specification cannot be proven data blindly).

Overview

1. Data-Oblivious Analysis

2. Gathered Results for PSF-Extensions

Results for flat, and pure$^+$ spec's

Theorem

For flat stream spec's data-oblivious productivity is decidable.

Hence there is a computable, data-obliviously optimal criterion for productivity of flat stream specifications.

Proposition

For pure$^+$ stream spec's: productivity = data-oblivious productivity.

Theorem

Productivity for pure$^+$ stream specifications is decidable.

Results for flat, and pure$^+$ spec's

Theorem

For flat stream spec's data-oblivious productivity is decidable.

Hence there is a computable, data-obliviously optimal criterion for productivity of flat stream specifications.

Proposition

For pure$^+$ stream spec's: productivity = data-oblivious productivity.

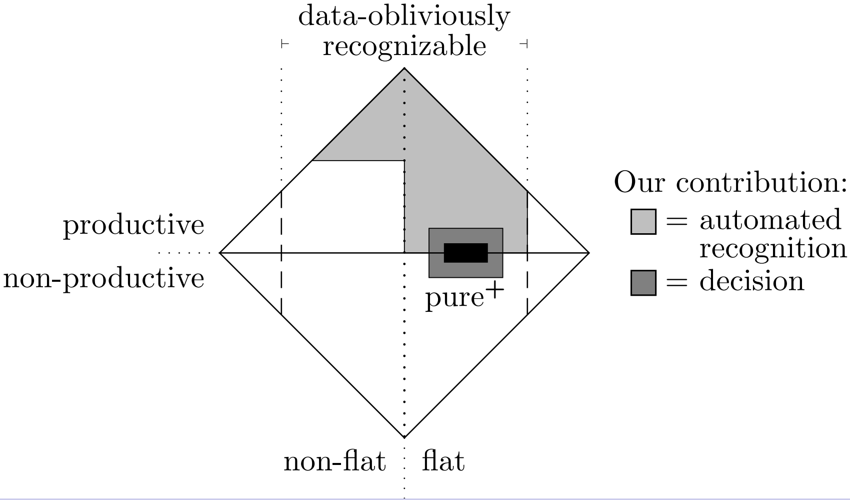
Theorem

Productivity for pure$^+$ stream specifications is decidable.

Deciding D-O Productivity

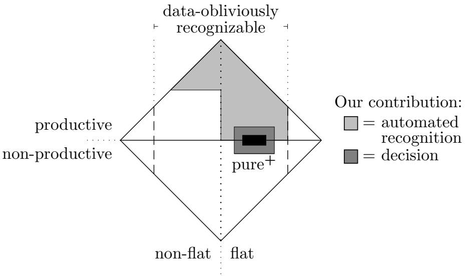
1. Input: a flat stream specification $\mathcal{R}$.
2. Stream function translation: for the stream functions f in $\mathcal{R}$, compute their d-o lower bounds $[\mathsf{f}] : \overline{\mathbb{N}} \to \overline{\mathbb{N}}$ (periodically increasing functions).
3. Stream constant translation: using (2), translate the root M_0 of $\mathcal{R}$ into a production term $[\mathsf{M}_0]$.
4. Production calculation: compute the production $\Pi([\mathsf{M}_0])$ of $[\mathsf{M}_0]$ in a production calculus (by a confluent, terminating TRS).
5. Decision taking: if $\Pi([\mathsf{M}_0]) = \infty$ then $\mathcal{R}$ is d-o productive, else $\mathcal{R}$ is not d-o productive.

Map of Stream Specifications



Results for PSF-extensions

1. for flat stream spec's:
   a decision method for data-oblivious productivity, yielding a computable, data-obliviously optimal criterion for productivity
2. for pure$^+$ stream spec's: a decision method for productivity
3. for friendly nesting stream spec's: a computable criterion for productivity
4. a productivity prover *ProPro* automating (1), (2), and (3)



References

Jörg Endrullis, Clemens Grabmayer, and Dimitri Hendriks.
Data-Oblivious Stream Productivity.
In *LPAR*, number 5330 in LNCS, pages 79–96. Springer, 2008.

Jörg Endrullis, Clemens Grabmayer, and Dimitri Hendriks.
ProPro: an Automated Productivity Prover.
`http://infinity.few.vu.nl/productivity/`, 2008.

Jörg Endrullis, Clemens Grabmayer, Dimitri Hendriks, Ariya Isihara, and Jan Willem Klop.
Productivity of Stream Definitions.
In *FCT*, number 4639 in LNCS, pages 274–287. Springer, 2007.