universität innsbruck

Cichon's Conjecture on the Slow Growing Hierarchy

The unexpected power of a pointwise hierarchy

Georg Moser

https://tcs-informatik.uibk.ac.at/

Motivation

Cichon's Conjecture

The derivational complexity induced by any termination order of order type α is bounded by the slow-growing hierarchy indexed by α.

1. conceptually, this conjecture links
   - logical complexities of a termination proof and
   - computational complexities of a given program
2. practically, this links diverse areas like
   - programming languages,
   - program analysis and
   - proof theory

the first part, should be conceived in the context of the following (far-reaching) result

> *Given any arithmetical theory T with proof-theoretic ordinal $\|T\|$, then the provable recursive functions of T are exactly those functions computable within complexity bounds by the Hardy functions H_α, $\alpha < \|T\|$.*

Course Schedule

Monday	(Universal) Termination of Term Rewrite Systems	17:30–18:20
Tuesday	The Slow-Growing Hierachy and Friends	9:00–9:50
Wednesday	Cichon's Conjecture and Counterexample	10:30–11:20

universität
innsbruck

(Universal) Termination of Term Rewrite Systems

Content

- terms and positions
- term rewrite systems
- termination of TRSs
- simplification orders
  - LPO
  - MPO
  - KBO
- well-founded monotone algebras
- (simple termination)

Example (Crash Course in Term Rewrite Systems)

- **signature** $\quad 0$ constant $\quad \mathsf{S}$ unary $\quad + \quad \times$ binary

- **rewrite rules**

$$\left.\begin{aligned} 0 + x &\to x \\ \mathsf{S}(x) + y &\to \mathsf{S}(x+y) \\ 0 \times x &\to 0 \\ \mathsf{S}(x) \times y &\to x \times y + y \end{aligned}\right\} \quad \text{TRS}$$

- **rewriting**

$$\begin{aligned} & \mathsf{S}(0) + \mathsf{S}(\mathsf{S}(0) \times \mathsf{S}(\mathsf{S}(0))) \\ \to\ & \mathsf{S}(0) + \mathsf{S}(0 \times \mathsf{S}(\mathsf{S}(0)) + \mathsf{S}(\mathsf{S}(0))) \\ \to\ & \mathsf{S}(0) + \mathsf{S}(0 + \mathsf{S}(\mathsf{S}(0))) \\ \to\ & \mathsf{S}(0) + \mathsf{S}(\mathsf{S}(\mathsf{S}(0))) \\ \to\ & \mathsf{S}(0 + \mathsf{S}(\mathsf{S}(\mathsf{S}(0)))) \\ \to\ & \mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{S}(0)))) \qquad \text{normal form} \end{aligned}$$

Example

- signature $0, 1, \ldots 9$ constants $+, :$ binary
- rewrite rules

$$\begin{array}{llll}
0+0 \to 0 & 1+0 \to 1 & \cdots & 9+0 \to 9 \\
0+1 \to 1 & 1+1 \to 2 & \cdots & 9+1 \to 1:0 \\
0+2 \to 2 & 1+2 \to 3 & \cdots & 9+2 \to 1:1 \\
0+3 \to 3 & 1+3 \to 4 & \cdots & 9+3 \to 1:2 \\
 & & \cdots & \\
0+7 \to 7 & 1+7 \to 8 & \cdots & 9+7 \to 1:6 \\
0+8 \to 8 & 1+8 \to 9 & \cdots & 9+8 \to 1:7 \\
0+9 \to 9 & 1+9 \to 1:0 & \cdots & 9+9 \to 1:8
\end{array}$$

$$\begin{array}{ll}
x + (y : z) \to y : (x+z) & 0 : x \to x \\
(x : y) + z \to x : (y+z) & x : (y : z) \to (x+y) : z
\end{array}$$

- rewriting $(2:3)+(7:7) \to 7:(2:3)+7$

$$\begin{array}{lll}
\to 7:(2:(3+7)) & \to 7:(2:(1:0)) & \to 7:((2+1):0) \\
\to 7:(3:0) & \to (7+3):0 & \to (1:0):0
\end{array}$$

Example

- **signature** $0, \mathsf{fib}$ constants S unary $\mathsf{f}, +, :$ binary
- **rules**

$$
\begin{array}{rclrcl}
0 + y & \to & y & \mathsf{fib} & \to & \mathsf{f}(\mathsf{S}(0), \mathsf{S}(0)) \\
\mathsf{S}(x) + y & \to & \mathsf{S}(x + y) & \mathsf{f}(x, y) & \to & x : \mathsf{f}(y, x + y)
\end{array}
$$

- **rewriting**

$$
\begin{array}{rcl}
\mathsf{fib} & \to & \mathsf{f}(\mathsf{S}(0), \mathsf{S}(0)) \\
& \to & \mathsf{S}(0) : \mathsf{f}(\mathsf{S}(0), \mathsf{S}(0) + \mathsf{S}(0)) \\
& \to & \mathsf{S}(0) : \mathsf{f}(\mathsf{S}(0), \mathsf{S}(0 + \mathsf{S}(0))) \\
& \to & \mathsf{S}(0) : \mathsf{f}(\mathsf{S}(0), \mathsf{S}(\mathsf{S}(0))) \\
& \to & \mathsf{S}(0) : \mathsf{S}(0) : \mathsf{f}(\mathsf{S}(\mathsf{S}(0)), \mathsf{S}(0) + \mathsf{S}(\mathsf{S}(0))) \\
& \to^+ & \mathsf{S}(0) : \mathsf{S}(0) : \mathsf{f}(\mathsf{S}(\mathsf{S}(0)), \mathsf{S}(\mathsf{S}(\mathsf{S}(0)))) \\
& \to^+ & \mathsf{S}(0) : \mathsf{S}(0) : \mathsf{S}^2(0) : \mathsf{f}(\mathsf{S}^3(0), \mathsf{S}^5(0)) \\
& \to^+ & \mathsf{S}(0) : \mathsf{S}(0) : \mathsf{S}^2(0) : \mathsf{S}^3(0) : \mathsf{f}(\mathsf{S}^5(0), \mathsf{S}^8(0))
\end{array}
$$

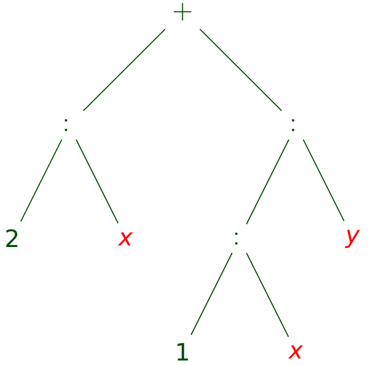
infinite computation

Definitions (Terms et al.)

- signature $\mathcal{F}$ function symbols with arities
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$ infinitely many
- terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$
- ground terms $\mathcal{T}(\mathcal{F})$

Operations

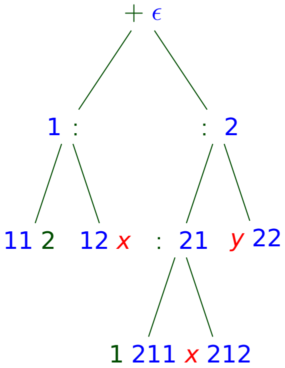
- $\mathsf{Var}(t)$ x y
- $\mathcal{F}\mathsf{un}(t)$ 1 2 $:$ $+$
- $\mathsf{rt}(t)$ $+$



Subterms and Positions

Definitions

- $s \trianglelefteq t$ ... s is subterm of t
- $t|_p$ ... take subterm of t at position p
- $t[s]_p$ ... replace subterm in t at position p by s
- $\mathrm{Pos}(t) = \mathrm{Pos}_{\mathcal{F}}(t) \cup \mathrm{Pos}_{\mathcal{V}}(t)$
- $p \leqslant q$ ... above
- $p \parallel q$ ... parallel



Substitutions

Definitions

- substitution is mapping $\sigma\colon \mathcal{V} \to \mathcal{T}(\mathcal{F},\mathcal{V})$ such that

$$\mathcal{D}\mathrm{om}(\sigma) = \underbrace{\{x \in \mathcal{V} \mid \sigma(x) \neq x\}}_{\text{domain}}$$

  is finite

- application of substitution σ to term t:

$$t\sigma = \begin{cases} \sigma(t) & \text{if } t \text{ is variable} \\ f(t_1\sigma,\dots,t_n\sigma) & \text{if } t = f(t_1,\dots,t_n) \end{cases}$$

- empty substitution ε $\quad (\mathcal{D}\mathrm{om}(\varepsilon) = \varnothing)$

Definitions (Term Rewriting Systems)

- rewrite rule ($l \to r$) is pair of terms l, r such that
  1. $l \notin \mathcal{V}$
  2. $\mathrm{Var}(r) \subseteq \mathrm{Var}(l)$
- term rewrite system (TRS) is pair $(\mathcal{F}, \mathcal{R})$
  1. $\mathcal{F}$ signature
  2. $\mathcal{R}$ set of rewrite rules between terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$
- binary relation $\to_{\mathcal{R}}$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ for every TRS $(\mathcal{F}, \mathcal{R})$:

$$s \to_{\mathcal{R}} t \quad \text{if} \quad \begin{array}{l} \exists\, p \in \mathrm{Pos}(s) \\ \exists\, l \to r \in \mathcal{R} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{rcll} s|_p & = & l\sigma & \quad \text{redex} \\ t & = & s[r\sigma]_p & \end{array}$$

How to Check for Termination

(more precisely *uniform* termination)

Definition

TRS is terminating if there are no infinite rewrite sequences (starting with any term)

Theorem

TRS $\mathcal{R}$ is terminating iff $\exists$ well-founded order $>$ on terms such that

$$s \to_{\mathcal{R}} t \quad \Longrightarrow \quad s > t$$

NB: inconvenient to check all rewrite steps

Fact

of course, (uniform) termination is an undecidable problem, more precisely it is Π^0_2-complete in the arithmetical hierarchy

Theorem

TRS $\mathcal{R}$ is terminating iff $\exists$ well-founded order $>$ on terms such that

1. $l \to r \in \mathcal{R} \implies l > r$
2. *$>$ is closed under contexts* $(s > t \;\Rightarrow\; C[s]_p > C[t]_p)$
3. *$>$ is closed under substitutions* $(s > t \;\Rightarrow\; s\sigma > t\sigma)$

Definition

binary relation $>$ on terms is reduction order if

1. closed under contexts
2. closed under substitutions
3. proper order (irreflexive and transitive)
4. well-founded

Definition

TRS $\mathcal{R}$ and $>$ are compatible if $l > r$ for all $l \to r \in \mathcal{R}$

Theorem

TRS $\mathcal{R}$ is terminating iff compatible with reduction order

Question

How to construct reduction orders ?

Answer

- use algebras (semantics)
- use induction (syntax)

Lexicographic Path Orders (LPO for short)

a syntactic method

Definition

- precedence is proper order $>$ on $\mathcal{F}$
- relation $>_{\mathrm{lpo}}$ (lexicographic path order) on terms:
  $s >_{\mathrm{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either
  1. $\exists\, i\ s_i >_{\mathrm{lpo}} t$ or $s_i = t$,
  2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j\ s >_{\mathrm{lpo}} t_j$, or
  3. $t = f(t_1, \ldots, t_n)$ and $\exists\, i$

$$\forall\, j \in [1, i-1]\ s_j = t_j \qquad s_i >_{\mathrm{lpo}} t_i \qquad \forall\, j > i\ s >_{\mathrm{lpo}} t_j$$

Theorem

$>_{\mathrm{lpo}}$ *is reduction order if* $>$ *is well-founded*

Example

$$\begin{aligned} x + 0 &\to x \\ x + \mathsf{S}(y) &\to \mathsf{S}(x+y) \\ x \times 0 &\to 0 \\ x \times \mathsf{S}(y) &\to x \times y + x \end{aligned} \qquad \times > + > \mathsf{S}$$

Theorem

- *if $> \subseteq \sqsupset$ then $>_{\mathsf{lpo}} \subseteq \sqsupset_{\mathsf{lpo}}$* (*incrementality*)
- *if $>$ is total then $>_{\mathsf{lpo}}$ is total on ground terms* (*well-order*)
- *following two problems are decidable:*

1 *instance:* *terms s, t $>$*

question: *$s >_{\mathsf{lpo}} t$?*

2 *instance:* *terms s, t*

question: *$\exists$ precedence $>$ such that $s >_{\mathsf{lpo}} t$?*

Ackermann Function and Lexicographic Path Order

Example

$$
\begin{aligned}
\mathsf{ack}(0,0) &\to 0 \\
\mathsf{ack}(0,\mathsf{S}(y)) &\to \mathsf{S}(\mathsf{S}(y)) \\
\mathsf{ack}(\mathsf{S}(x),0) &\to \mathsf{ack}(x,\mathsf{S}(0)) \\
\mathsf{ack}(\mathsf{S}(x),\mathsf{S}(y)) &\to \mathsf{ack}(x,\mathsf{ack}(\mathsf{S}(x),y))
\end{aligned}
\qquad \mathsf{ack} > \mathsf{S}
$$

Remark

LPO can handle multiple-recursive functions

Definition

- precedence is proper order $>$ on $\mathcal{F}$
- relation $>_{\mathsf{mpo}}$ (multiset path order) on terms:
  $s >_{\mathsf{mpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either
  1. $\exists\, i\ s_i >_{\mathsf{mpo}} t$ or $s_i = t$
  2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j\ s >_{\mathsf{mpo}} t_j$
  3. $t = f(t_1, \ldots, t_n)$ and $\{s_1, \ldots, s_n\} >_{\mathsf{mpo}}^{\mathsf{mul}} \{t_1, \ldots, t_n\}$
-

$$M >_{\mathsf{mpo}}^{\mathsf{mul}} N \iff \overbrace{M - N}^{\text{multiset difference}} \neq \varnothing \quad \wedge$$
$$\forall\, t \in N - M\ \exists\, s \in M - N \quad s >_{\mathsf{mpo}} t$$

Theorem

$>_{\mathsf{mpo}}$ *is reduction order if* $>$ *is well-founded*

Definition

- weight function (w, w_0) consists of mapping $\mathrm{w}\colon \mathcal{F} \to \mathbb{N}$ and constant $w_0 > 0$ such that $\mathrm{w}(c) \geqslant w_0$ for all constants $c \in \mathcal{F}$
- weight of term t is

$$\mathrm{w}(t) = w_0 \cdot \Big(\sum_{x \in \mathcal{V}\mathrm{ar}(t)} |t|_x \Big) + \sum_{f \in \mathcal{F}\mathrm{un}(t)} \mathrm{w}(f) \cdot |t|_f$$

- weight function (w, w_0) is admissible for precedence $>$ if

$$f > g \text{ for all } g \in \mathcal{F} \setminus \{f\}$$

whenever f is unary function symbol in $\mathcal{F}$ with $w(f) = 0$

Example

$$\mathrm{w}(\circ) = \mathrm{w}(\mathsf{S}) = 0 \quad \mathrm{w}(0) = 1 \qquad \mathsf{S} > \circ > 0$$

Definition

- precedence is proper order $>$ on $\mathcal{F}$
- admissible weight function (w, w_0)
- relation $>_{\mathrm{kbo}}$ (Knuth-Bendix order) on terms:
  $s >_{\mathrm{kbo}} t$ if $|s|_x \geqslant |t|_x$ for all $x \in \mathcal{V}$ and either
  1. $w(s) > w(t)$,
  2. $w(s) = w(t)$ and either
     ❶ $\exists\, n > 0 \;\exists\, x \in \mathcal{V}\; s = f^n(x)$ and $t = x$
     ❷ $s = f(s_1, \dots, s_n)$ and $t = f(t_1, \dots, t_n)$ and $\exists\, i$

     $$\forall\, j < i\; s_j = t_j \qquad s_i >_{\mathrm{kbo}} t_i$$

     ❸ $s = f(s_1, \dots, s_n)$ and $t = g(t_1, \dots, t_m)$ and $f > g$

Theorem

$>_{\mathrm{kbo}}$ is reduction order if $>$ is well-founded and (w, w_0) admissible

Theorem

- *if* $>\ \subseteq\ \sqsupset$ *and* (w, w_0) *admissible then* $>_{\mathsf{kbo}}\ \subseteq\ \sqsupset_{\mathsf{kbo}}$ *(incrementality)*
- *if* $>$ *is total then* $>_{\mathsf{kbo}}$ *is total on ground terms* *(well-order)*
- *following two problems are decidable:*

1 *instance:* *terms s, t* $\quad >\quad (\mathsf{w}, w_0)$

question: $s >_{\mathsf{kbo}} t$ *?*

2 *instance:* *terms s, t*

question: $\exists$ *precedence* $>$ *and admissible* (w, w_0) *such that* $s >_{\mathsf{kbo}} t$ *?*

Example

$$\begin{array}{lcl} \mathsf{g}(\mathsf{g}(x)) & \to & \mathsf{f}(x) \\ \mathsf{f}(\mathsf{g}(x)) & \to & \mathsf{g}(\mathsf{f}(x)) \end{array} \qquad \mathsf{f} > \mathsf{g} \wedge \mathsf{w}(\mathsf{f}) = \mathsf{w}(\mathsf{g}) = 1$$

Well-Founded Monotone Algebras

Definitions

- well-founded monotone $\mathcal{F}$-algebra (WFMA) $(\mathcal{A}, >)$ is non-empty algebra $\mathcal{A} = (A, \{f_\mathcal{A}\}_{f \in \mathcal{F}})$ together with well-founded order $>$ on A such that every $f_\mathcal{A}$ is strictly monotone in all coordinates:

$$f_\mathcal{A}(a_1, \ldots, a_i, \ldots, a_n) \;>\; f_\mathcal{A}(a_1, \ldots, b, \ldots, a_n)$$

  for all $a_1, \ldots, a_n, b \in A$ and $i \in [1, n]$ with $a_i > b$
- binary relation $>_\mathcal{A}$ on terms:

$$s >_\mathcal{A} t \quad \text{if} \quad \underbrace{[\alpha]_\mathcal{A}(s)}_{\text{interpretation of } s \text{ in } \mathcal{A} \text{ under assignment } \alpha} > [\alpha]_\mathcal{A}(t) \quad \text{for all assignments } \alpha$$

- TRS $\mathcal{R}$ and WFMA $(\mathcal{A}, >)$ are compatible if $\mathcal{R}$ and $>_\mathcal{A}$ are compatible

Completeness of Well-founded Monotone Algebras

a semantic method

Theorem

- *$>_{\mathcal{A}}$ is reduction order for every WFMA $(\mathcal{A}, >)$*
- *TRS is terminating iff compatible with WFMA*

Definition

TRS $\mathcal{R}$ is polynomially terminating if compatible with WFMA $(\mathcal{A}, >)$ such that

1. carrier of $\mathcal{A}$ is $\mathbb{N}$
2. $>$ is standard order on $\mathbb{N}$
3. $f_{\mathcal{A}}$ is polynomial for every f

Example

$$
\begin{aligned}
x + 0 &\to x & 0_{\mathcal{A}} &:= 1 \\
x + \mathsf{S}(y) &\to \mathsf{S}(x + y) & \mathsf{S}_{\mathcal{A}} &:= \lambda x \,.\, x + 1 \\
x \times 0 &\to 0 & +_{\mathcal{A}} &:= \lambda xy \,.\, x + 2y \\
x \times \mathsf{S}(y) &\to x \times y + x & \times_{\mathcal{A}} &:= \lambda xy \,.\, (x+1)(y+1)^2
\end{aligned}
$$

Remark

without further restrictions, polynomially terminating TRS surpass polynomial functions

Further Reading

Franz Baader and Tobias Nipkow.
Term Rewriting and All That
Cambridge University Press, 1998

TeReSe.
Term Rewriting Systems
Cambridge Tracts in Theoretical Computer Science, volume 55, 2003

Enno Ohlebusch.
Advanced Topics in Term Rewriting
Springer, 2002

Hoon Hong and Dalibor Jakuš.
Testing Positiveness of Polynomials
JAR 21(1), pp. 23 – 38, 2004

J. Endrullis, J. Waldmann, and H. Zantema.
Matrix Interpretations for Proving Termination of Term Rewriting.
JAR, 40(3):195–220, 2008.

T. Arts and J. Giesl.
Termination of Term Rewriting using Dependency Pairs.
TCS, 236(1,2):133–178, 2000.

N. Hirokawa and A. Middeldorp.
Tyrolean Termination Tool: Techniques and Features.
IC, 205:474–511, 2007.

Jürgen Giesl, René Thiemann, Peter Schneider-Kamp and Stephan Falke.
Mechanizing and Improving Dependency Pairs
JAR 37(3), pp. 155 – 203, 2006

R. Thiemann.
The DP Framework for Proving Termination of Term Rewriting.
PhD thesis, University of Aachen, 2007.

universität
innsbruck

Thank You for Your Attention!