Constructing Lower Bounds on the Derivational Complexity of Rewrite Systems

Dieter Hofbauer, BA Nordhessen, Germany
Johannes Waldmann, HTWK Leipzig, Germany

Derivational Complexity: Definition

The *derivation height* of term t modulo system R is the maximal length of an R-derivation starting in t:

$$\boxed{\mathrm{dh}_R(t) = \max\{n \mid \exists s : t \to_R^n s\}}$$

The *derivational complexity* of R maps natural number n to the maximal derivation height of terms of size at most n:

$$\boxed{\mathrm{dc}_R(n) = \max\{\mathrm{dh}_R(t) \mid \mathrm{size}(t) \leq n\}}$$

This is a worst case complexity measure.

How about the following systems?

- $\{aab \to ba\}$, $\{ab \to ba\}$, $\{ab \to baa\}$, $\{aa \to aba\}$

Example: Bubble Sort

$$\boxed{ab \to ba}$$

- Upper bound $O(n^2)$ from the (matrix) interpretation

$$[a](x, y) = (x + y, y)$$
$$[b](x, y) = (x, y + 1)$$

$$\begin{aligned}[ab](x, y) &= (x + y + 1, y + 1) \\ &> (x + y, y + 1) = [ba](x, y)\end{aligned}$$

For each string w, $[w](0, 0) \leq (|w|^2, |w|)$.

- Lower bound $\Omega(n^2)$ from the family of derivations

$$a^n b^n \to_R^{n^2} b^n a^n$$

Derivational Complexity: Exercises

Find lower bounds for the derivational complexity of

- $R_1 = \{ba \to acb,\ bc \to abb\}$
- $R_2 = \{ba \to acb,\ bc \to cbb\}$
- $R_3 = \{ba \to aab,\ bc \to cbb\}$

Hint: one system is doubly exponential, one is multiply exponential, one is non-terminating.

A lower bound is proven by presenting a family of derivations that achieves the desired length.

Research Program

- Deduce upper bounds on the derivational complexity from termination proofs.
- Characterize complexity classes via termination proof methods: *Implicit Computational Complexity*.
- This talk:

  Deduce lower bounds on the derivational complexity from derivation patterns.

  Applications:

  - "debugging" of rewrite systems
  - evaluating the strength of the automated methods for finding upper bounds (complexity category of the termination competition)

www.termination-portal.org

- Workshop on termination (1st WST'93 – 11th WST'10)
- Termination competition ('04 – '10)
- Problems
  termination problem data base (tpdb) at
  `termcomp.uibk.ac.at/status/downloads/`
- Tools (provers, verifiers)
- Complexity category, since '08
  - CaT [Korp, Sternagel, Zankl]
  - TCT [Avanzini, Moser, Schnabl]
  - Matchbox [W]

  Focus up to now: (polynomial) upper bounds

  This talk: lower bounds

Upper / Lower Bounds: Examples

1. $R = \{aa \to aba\}$, $\mathrm{dc}_R \in \Theta(n)$

Upper / Lower Bounds: Examples

1. $R = \{aa \to aba\}$, $\mathrm{dc}_R \in \Theta(n)$
2. $R = \{ab \to ba\}$, $\mathrm{dc}_R \in \Theta(n^2)$

Upper / Lower Bounds: Examples

1. $R = \{aa \to aba\}$, $\mathrm{dc}_R \in \Theta(n)$
2. $R = \{ab \to ba\}$, $\mathrm{dc}_R \in \Theta(n^2)$
3. $R = \{ab \to baa\}$, $\mathrm{dc}_R \in \Theta(2^n)$

Upper / Lower Bounds: Examples

1. $R = \{aa \to aba\}$, $\mathrm{dc}_R \in \Theta(n)$
2. $R = \{ab \to ba\}$, $\mathrm{dc}_R \in \Theta(n^2)$
3. $R = \{ab \to baa\}$, $\mathrm{dc}_R \in \Theta(2^n)$
4. $R = \{aabab \to aPb, aP \to PAa, aA \to Aa,$
   $bP \to bQ, QA \to aQ, Qa \to babaa\}$
   dc_R not primitive recursive (Ackermann)

Upper / Lower Bounds: Examples

1. $R = \{aa \to aba\}$, $\mathrm{dc}_R \in \Theta(n)$
2. $R = \{ab \to ba\}$, $\mathrm{dc}_R \in \Theta(n^2)$
3. $R = \{ab \to baa\}$, $\mathrm{dc}_R \in \Theta(2^n)$
4. $R = \{aabab \to aPb, aP \to PAa, aA \to Aa,$
   $bP \to bQ, QA \to aQ, Qa \to babaa\}$
   dc_R not primitive recursive (Ackermann)
5. Etc. (string rewriting is computationally complete)

Upper / Lower Bounds: Examples

1. $R = \{aa \to aba\}$, $\mathrm{dc}_R \in \Theta(n)$
2. $R = \{ab \to ba\}$, $\mathrm{dc}_R \in \Theta(n^2)$
3. $R = \{ab \to baa\}$, $\mathrm{dc}_R \in \Theta(2^n)$
4. $R = \{aabab \to aPb, aP \to PAa, aA \to Aa, bP \to bQ, QA \to aQ, Qa \to babaa\}$
   dc_R not primitive recursive (Ackermann)
5. Etc. (string rewriting is computationally complete)

We can deduce some of the upper bounds automatically:

1. via match bounds
2. via upper triangular 3×3 matrix interpretations
3. via matrix interpretations

Upper Bounds

- polynomial interpretations $\rightsquigarrow$ doubly exponential [Lautemann / Geupel / H / Zantema / ...]
- multiset path orders $\rightsquigarrow$ primitive recursive [H]
- lexicographic path orders $\rightsquigarrow$ multiple recursive [Weiermann]
- Knuth-Bendix orders $\rightsquigarrow$ multiple recursive (2-rec) [H, Lautemann / Touzet / Lepper / Bonfante / Moser]
- Related [Buchholz / Touzet / Weiermann / Moser ...]
- match bounds $\rightsquigarrow$ linear [Geser, H, W]
- matrix interpretations $\rightsquigarrow$ exponential [H, W]

Smaller Upper Bounds

Challenge: *Small* complexity classes.
Here, previous upper bound results heavily overestimate dc_R.

Some remedies:

- Syntactic restrictions of standard path orders
  - light multiset path order LMPO [Marion]
  - polynomial path order POP*: innermost derivations on constructor-based terms [Avanzini, Moser], cf. [Bellantoni, Cook]
- Matrix interpretations of particular shape [W]
- Context-dependent interpretations
  [H / Schnabl, Moser]

Lower Bound for Bubble Sort

$$\boxed{ab \to ba}$$

Rule: $ab \to^1 ba$

Compose: $a^2b \to^2 ba^2$

Generalize: $aa^nb \to^{n+1} baa^n$

Verify (induction step):

$$\begin{aligned} aa^{n+1}b &\sim aaa^nb \\ &\to^{n+1} abaa^n \\ &\to^1 baaa^n \\ &\sim baa^{n+1} \end{aligned}$$

Result: Linear lower bound

Bubble Sort (cont'd)

$$\boxed{ab \to ba}$$

Pattern: $aa^n b \to^{n+1} baa^n$

Compose: $aa^n bb \to^{2(n+1)} bbaa^n$

Generalize: $aa^n bb^m \to^{(m+1)(n+1)} bb^m aa^n$

Verify (induction step):

$$\begin{aligned} aa^n bb^{m+1} &\sim aa^n bb^m b \\ &\to^{(m+1)(n+1)} bb^m aa^n b \\ &\to^{n+1} bb^m baa^n \\ &\sim bb^{m+1} aa^n \end{aligned}$$

Result: Quadratic lower bound

Similar Example: Associativity

$$\boxed{f(f(x,y),z) \to f(x,f(y,z))}$$

- For $R = [f(x,\cdot)]$ and $L = [f(\cdot,z)]$,

$$L(R(y)) = f(f(x,y),z) \to f(x,f(y,z)) = R(L(y))$$

- Again,

$$L^n(R^m(y)) \to_R^{n\cdot m} R^n(L^m(y))$$

this still looks like string rewriting (on $\Sigma = \{L, R\}$)

Example: Real Terms

$$\boxed{f(s(x), y) \to f(x, s(y))}$$

Rule: $f(s(x), y) \to^1 f(x, s(y))$

Compose: $f(s^2(x), y) \to^2 f(x, s^2(y))$

Generalize: $f(s(s^n(x)), y) \to^{n+1} f(x, s(s^n(y)))$

Verify (induction step):

$$\begin{aligned} f(s(s^{n+1}(x)), y) &\sim f(s(s(s^n(x))), y) \\ &\to^1 f(s(s^n(x)), s(y)) \\ &\to^{n+1} f(x, s(s^n(s(y)))) \\ &\sim f(x, s(s^{n+1}(y))) \end{aligned}$$

Result: Linear lower bound

Example: Real Terms (cont'd)

$$\boxed{f(s(x), y) \to f(x, s(y)), \;\; s(f(x, y)) \to f(y, x)}$$

Rule: $s(f(x, y)) \to^{1} f(y, x)$

Compose: $s(f(s^{n+1}(x), y)) \to^{n+2} f(s^{n+1}(y), x)$

Compose: $s(s(f(s^{n+1}(x), y))) \to^{2(n+2)} f(s^{n+1}(x), y)$

Generalize: $s(s^{m}(f(s^{n+1}(x)), x) \to^{(m+1)(n+2)} f(s^{n+1}(x), x)$

Verify: similar to the previous example

Result: Quadratic lower bound

Derivation Patterns

derivation pattern consists of:

- lhs, rhs: term pattern
- length: numerical pattern (polynomial, ...)

term pattern constructed from:

- term variable
- function symbol with term patterns as arguments
- *iterated context application*, consisting of:
  - linear context: term with one hole
  - iteration count: (simple?) numerical pattern
  - argument: term pattern

pattern *compatible* with rewrite system R:
for any assignment of term and numerical variables, the instantiated pattern is an R-derivation of the given length.

Constructing Derivation Patterns

- rules are patterns
- compose patterns via overlap closures
- generalize via embedding
- verify by enumerating reachable terms
(apply verified patterns and induction hypothesis
modulo context equalities)

Context Equalities

expand top:	$C^{k+1}(t) \sim C(C^k(t))$
expand bottom:	$C^{k+1}(t) \sim C^k(C(t))$
remove:	$C^0(t) \sim t$
rotate:	$(CD)^k C(t) \sim C(DC)^k(t)$

Derivation Height of the Patterns

- avoid (symbolic) numerical calculations
- storing just the *degree* of the polynomial
- if induction hypthesis is used *once* in the verification of the induction step,
  then the degree of the inductive pattern is $1+$ max degree of other patterns used.
- needs extension if several numerical variables occur
- need to check that lhs of patterns have linear size
  this is enforced by syntactic restrictions (context is "term with hole", not "term pattern with hole")

Polynomials of higher Degree

our patterns can describe (some) polynomial length derivations of any given degree.

$$B_d = \{ki \to jk \mid k > i, j\} \text{ over } \Sigma_d = \{1, 2, \ldots, d\}$$

$B_2 = \{21 \to 12\}$, $B_3 = \{21 \to 12, 31 \to 23, 32 \to 13, \ldots\}$

- lower bound:
  for $d \geq 2$, we have $d^n \ldots 2^n 1^n \to^{\Theta(n^d)} 1^n 2^n \ldots d^n$
- upper bound:
  upper triangular matrix interpretation of dimension d

Some non-polynomial patterns

when searching for polynomial patterns,
may find something else along the way

- exponential patterns
  - iterate a linear function of slope > 1
  - use induction hypothesis more than once
- non-terminating patterns (looping, non-looping)
  - lhs of pattern is constant, but rhs is not

Example: Exponential Lower Bound

$$\boxed{ab \to baa}$$

Rule:	$ab \to^1 baa$
Compose:	$a^2b \to^2 ba^4$
Generalize:	$aa^nb \to^{n+1} ba^{2(n+1)}$

using the above, prove the $\Omega(2^n)$ lower bound pattern:

Rule:	$ab \to^1 baa$
Compose:	$ab^2 \to^3 b^2a^{2^2}$
Generalize:	$abb^n \to^{2^{n+1}-1} bb^na^{2^{n+1}}$

Exponential, for a Different Reason

$$\{0 \to 1, 1 \to C, 0C \to 10, 1C \to C0\}$$

- Pattern $00^k \to^{\geq 2^k} C0^k$.
- Base: $k \mapsto 0$ gives $00^0 = 0 \to^2 C = C0^0$
- Step: $k \mapsto k+1$ gives $00^{k+1} \to^{2^{k+1}} C0^{k+1}$.
  expand: 000^k, apply hypothesis: $0C0^k$, apply rule: 100^k, apply hypothesis: $1C0^k$, apply rule: $C00^k$, collect: $C0^{k+1}$.

exponential because induction hypthesis is applied twice in the induction step

Non-Termination

Infinite lower bound . . .
Simple forms of non-termination

- Cycles: $t \rightarrow_R^+ t$
- Loops: $t \rightarrow_R^+ C(t\sigma)$
- Self-Embedding Patterns,
  e.g., $ab^x dc \rightarrow^+ ab^{x+1} dc$ (Geser/Zantema, Oppelt)

our method should be able to find patterns for such derivations:
the lhs is constant (does not depend on numerical variables) while the length and/or rhs are not constant

Beyond Loops

Oppelt's tool `nonloop`

- overlap closures
- derivation patterns
- self-embedding patterns
- inference rules on patterns
- Expl.s from the database:
  `oppelt08/*` and `Zantema/z073`

Oppelt's nonloop (cont'd)

$$bc \to dc,\ bd \to db,\ ad \to abb$$

$$bd \to^+ db$$

$$b^x d \to^+ db^x$$

$$b^{x+1} d \to^+ db^{x+1}$$

$$b^{x+1} d \to^+ db^x b$$

$$b^{x+1} dc \to^+ db^x bc$$

$$b^{x+1} dc \to^+ db^x dc$$

$$ab^{x+1} dc \to^+ adb^x dc$$

$$ab^{x+1} dc \to^+ abbb^x dc$$

$$ab^{x+1} dc \to^+ ab^{x+2} dc$$

results in a self-embedding derivation pattern

Conclusion

- Rather restricted form of patterns:
  only one-place contexts, restricted nesting
- No proper higher-order unification
- But suffices for many examples
- Implementation is work in progress
  (main task is to control the search:
  keep (promising) patterns in priority queue)