Embracing Infinity – Termination of String Rewriting by Almost Linear Weight Functions

Dieter Hofbauer

ASW – BA Saarland

16th Workshop on Termination
Oxford, UK, July 18–19, 2018

Weight functions $w : \Sigma \to \mathbb{N}$

- Among the simplest methods for proving termination: Additively extended to $w : \Sigma^* \to \mathbb{N}$.
- If $w(\ell) > w(r)$ for each rule $\ell \to r$ in R, then R is terminating, and R has linear derivational complexity.

Example: $\{aba \to aa\}$

Example: $\{aa \to aba\}$

Weight functions $w : \Sigma \to \mathbb{N}$

- Among the simplest methods for proving termination: Additively extended to $w : \Sigma^* \to \mathbb{N}$.
- If $w(\ell) > w(r)$ for each rule $\ell \to r$ in R, then R is terminating, and R has linear derivational complexity.

Example: $\{aba \to aa\}$

- Choosing $w(a) = 0$ and $w(b) = 1$, we get $w(aba) = w(a) + w(b) + w(a) = 1 > 0 = w(a) + w(a) = w(aa)$.
- In general, $w(aba) > w(aa)$ for $w(b) > 0$.

Example: $\{aa \to aba\}$

Weight functions $w : \Sigma \to \mathbb{N}$

- Among the simplest methods for proving termination: Additively extended to $w : \Sigma^* \to \mathbb{N}$.
- If $w(\ell) > w(r)$ for each rule $\ell \to r$ in R, then R is terminating, and R has linear derivational complexity.

Example: $\{aba \to aa\}$

- Choosing $w(a) = 0$ and $w(b) = 1$, we get $w(aba) = w(a) + w(b) + w(a) = 1 > 0 = w(a) + w(a) = w(aa)$.
- In general, $w(aba) > w(aa)$ for $w(b) > 0$.

Example: $\{aa \to aba\}$

- No weight function with $w(aa) > w(aba)$ exists: $w(aa) - w(aba) = -w(b) \leq 0$.
- Below: Termination proof with generalized weight function.

Syllable decomposition of strings

Definition

For *split letter* $a \in \Sigma$ and $x_i \in (\Sigma \setminus \{a\})^*$

$$\mathrm{split}_a(x_1 a x_2 a \ldots a x_n) = x_1 \ x_2 \ \ldots \ x_n$$

Syllable decomposition of strings

Definition

For *split letter* $a \in \Sigma$ and $x_i \in (\Sigma \setminus \{a\})^* = \Gamma$

$$\mathrm{split}_a(x_1 a x_2 a \ldots a x_n) = (x_1, x_2, \ldots, x_n) \in \Gamma^+$$

Syllable decomposition of strings

Definition

For *split letter* $a \in \Sigma$ and $x_i \in (\Sigma \setminus \{a\})^* = \Gamma$

$$\text{split}_a(x_1 a x_2 a \ldots a x_n) = (x_1, x_2, \ldots, x_n) \in \Gamma^+$$

- The resulting alphabet Γ is infinite (except for $\Sigma = \{a\}$).
- This talk: Consider the simple case of a two-letter alphabet, $\Sigma = \{a, b\}$ say. Then an element of $\Gamma = \{b^n \mid n \in \mathbb{N}\}$ is uniquely determined by its length.

Syllable decomposition of strings

Definition

For *split letter a* $\in \Sigma$ and $x_i \in (\Sigma \setminus \{a\})^* = \Gamma$

$$\mathrm{split}_a(x_1 a x_2 a \dots a x_n) = (x_1, x_2, \dots, x_n) \in \Gamma^+$$

- The resulting alphabet Γ is infinite (except for $\Sigma = \{a\}$).
- This talk: Consider the simple case of a two-letter alphabet, $\Sigma = \{a, b\}$ say. Then an element of $\Gamma = \{b^n \mid n \in \mathbb{N}\}$ is uniquely determined by its length.

Definition

For $x \in \Sigma^*$ and $a \in \Sigma$ with $\mathrm{split}_a(x) = (x_1, \dots, x_n)$

$$\mathrm{slit}_a(x) = (|x_1|, \dots, |x_n|) \in \mathbb{N}^+$$

Syllable decomposition (cont'd)

- Used in Sakai'84 for defining the Kachinuki order, which coincides on strings with RPO from Dershowitz'82:
  The syllable decomposition is recursively continued.
- This talk: Decomposition only as a single first step.

Syllable decomposition (cont'd)

- Used in Sakai'84 for defining the Kachinuki order, which coincides on strings with RPO from Dershowitz'82: The syllable decomposition is recursively continued.
- This talk: Decomposition only as a single first step.

Extend the natural bijection between Γ and $\mathbb{N}$ to rewriting

- For system R over Σ define system $\mathrm{slit}_a(R)$ over $\mathbb{N}$ so that $\mathrm{slit}_a : \Sigma^* \to \mathbb{N}^+$ is an isomorphism between the relational structures $(\Sigma^*, \to_R)$ and $(\mathbb{N}^+, \to_{\mathrm{slit}_a(R)})$.
- Then termination of R is equivalent to termination of $\mathrm{slit}_a(R)$.
- Just as the alphabet $\mathbb{N}$, the system $\mathrm{slit}_a(R)$ is infinite: Properly handle contexts of rule application.

Syllable decompositon: Properly handle contexts

Definition (draft)

$\mathsf{slit}_a(\ell \to r) = (\mathsf{slit}_a(\ell) \to \mathsf{slit}_a(r))$, and $\mathsf{slit}_a(R)$ accordingly.

Syllable decompositon: Properly handle contexts

Definition (draft)

$\mathrm{slit}_a(\ell \to r) = (\mathrm{slit}_a(\ell) \to \mathrm{slit}_a(r))$, and $\mathrm{slit}_a(R)$ accordingly.

Example: $\{aa \to aba\}$

- $\mathrm{split}_a(R) = \{(\epsilon, \epsilon, \epsilon) \to (\epsilon, b, \epsilon)\}$, thus $\mathrm{slit}_a(R) = \{(0, 0, 0) \to (0, 1, 0)\}$.
- Isomorphism broken: R is applicable to $baab$, but $\mathrm{slit}_a(R)$ is not applicable to $\mathrm{slit}_a(baab) = (1, 0, 1)$.

Syllable decompositon: Properly handle contexts

Definition (draft)

$\text{slit}_a(\ell \to r) = (\text{slit}_a(\ell) \to \text{slit}_a(r))$, and $\text{slit}_a(R)$ accordingly.

Example: $\{aa \to aba\}$

- $\text{split}_a(R) = \{(\epsilon, \epsilon, \epsilon) \to (\epsilon, b, \epsilon)\}$, thus $\text{slit}_a(R) = \{(0, 0, 0) \to (0, 1, 0)\}$.
- Isomorphism broken: R is applicable to $baab$, but $\text{slit}_a(R)$ is not applicable to $\text{slit}_a(baab) = (1, 0, 1)$.

Definition

$\text{slit}_a(\ell \to r) = \{p + \text{slit}_a(\ell) + q \to p + \text{slit}_a(r) + q \mid p, q \in \mathbb{N}\}$

Overloaded notation:

$$p + (k_1, \ldots, k_n) + q = (p + k_1, \ldots, k_n + q)$$

Syllable decompositon: Properly handle contexts

Definition (draft)

$\mathrm{slit}_a(\ell \to r) = (\mathrm{slit}_a(\ell) \to \mathrm{slit}_a(r))$, and $\mathrm{slit}_a(R)$ accordingly.

Example: $\{aa \to aba\}$

- $\mathrm{split}_a(R) = \{(\epsilon, \epsilon, \epsilon) \to (\epsilon, b, \epsilon)\}$, thus $\mathrm{slit}_a(R) = \{(0, 0, 0) \to (0, 1, 0)\}$.
- Isomorphism broken: R is applicable to $baab$, but $\mathrm{slit}_a(R)$ is not applicable to $\mathrm{slit}_a(baab) = (1, 0, 1)$.

Definition

$\mathrm{slit}_a(\ell \to r) = \{p + \mathrm{slit}_a(\ell) + q \to p + \mathrm{slit}_a(r) + q \mid p, q \in \mathbb{N}\}$

Example: $\{aa \to aba\}$

- $\mathrm{slit}_a(R) = \{(p, 0, q) \to (p, 1, q) \mid p, q \in \mathbb{N}\}$
- $\mathrm{slit}_b(R) = \{(p + 2 + q) \to (p + 1, 1 + q) \mid p, q \in \mathbb{N}\}$

Infinite alphabets and infinite rewrite systems

Theorem

For an alphabet Σ of size two and $a \in \Sigma$, termination of R over Σ and termination of $\mathsf{slit}_a(R)$ over $\mathbb{N}$ are equivalent.

Infinite alphabets and infinite rewrite systems

Theorem

For an alphabet Σ of size two and $a \in \Sigma$, termination of R over Σ and termination of $\mathrm{slit}_a(R)$ over $\mathbb{N}$ are equivalent.

Example: $R = \{aa \to aba\}$

- No weight function proves termination of R.
- Instead, consider $S = \mathrm{slit}_a(R) = \{(p, 0, q) \to (p, 1, q)\}$ and $w : \mathbb{N} \to \mathbb{N}$ where $w(0) = 1$ and $w(n) = 0$ for $n \neq 0$.
- Note that $\mathbb{N}$ is the alphabet of S, thus the domain of w, and also the codomain of w as a weight function.
- $w((p, 0, q)) > w((p, 1, q))$,
  so S is terminating, so R is terminating.

Finite proof obligations: Almost linear functions

- Termination by weight functions applies to infinite alphabets and infinite systems.
- To obtain both finite proof obligations and finite proof certificates: Restricted classes of functions
- This talk: Linear functions with finitely many exceptions.

Finite proof obligations: Almost linear functions

- Termination by weight functions applies to infinite alphabets and infinite systems.
- To obtain both finite proof obligations and finite proof certificates: Restricted classes of functions
- This talk: Linear functions with finitely many exceptions.

Definition

An *almost linear function* is the union of a function $e : E \to \mathbb{Z}$ with finite domain $E \subseteq \mathbb{Z}$ and a linear function $h : \mathbb{Z} \setminus E \to \mathbb{Z}$ with $h(n) = an + b$ for $a, b \in \mathbb{Z}$.

- Short-hand notation for $e = \{(p_1, q_1), \ldots, (p_k, q_k)\}$:
  $w : p_1 \mapsto q_1, \ldots, p_k \mapsto q_k$, else $n \mapsto an + b$

Almost linear functions (cont'd)

Example: $R = \{aa \to aba\}$

- For $\mathrm{slit}_a(R)$ use $w : 0 \mapsto 1$, else $n \mapsto 0$.

Almost linear functions (cont'd)

Example: $R = \{aa \to aba\}$

- For $\text{slit}_a(R)$ use $w : 0 \mapsto 1$, else $n \mapsto 0$.

Example: $R = \{aa \to aba\}$

- For $\text{slit}_b(R) = \{(p + 2 + q) \to (p + 1, 1 + q) \mid p, q \in \mathbb{N}\}$ use $w : 0 \mapsto 0$, else $n \mapsto n - 1$: Easy, no case distinction:
  $w(p+2+q) = p+2+q-1 > p+q = w(p+1)+w(1+q)$.

Almost linear functions (cont'd)

Example: $R = \{aa \to aba\}$

- For $\mathrm{slit}_a(R)$ use $w : 0 \mapsto 1$, else $n \mapsto 0$.

Example: $R = \{aa \to aba\}$

- For $\mathrm{slit}_b(R) = \{(p+2+q) \to (p+1, 1+q) \mid p, q \in \mathbb{N}\}$ use $w : 0 \mapsto 0$, else $n \mapsto n-1$: Easy, no case distinction: $w(p+2+q) = p+2+q-1 > p+q = w(p+1) + w(1+q)$.

To be suitable as a weight function,
w has to be *non-negative on $\mathbb{N}$*, i. e., $w(\mathbb{N}) \subseteq \mathbb{N}$.

Almost linear functions (cont'd)

Example: $R = \{aa \to aba\}$

- For $\mathrm{slit}_a(R)$ use $w : 0 \mapsto 1$, else $n \mapsto 0$.

Example: $R = \{aa \to aba\}$

- For $\mathrm{slit}_b(R) = \{(p + 2 + q) \to (p + 1, 1 + q) \mid p, q \in \mathbb{N}\}$ use $w : 0 \mapsto 0$, else $n \mapsto n - 1$: Easy, no case distinction: $w(p+2+q) = p+2+q-1 > p+q = w(p+1)+w(1+q)$.

To be suitable as a weight function,
w has to be *non-negative on* $\mathbb{N}$, i. e., $w(\mathbb{N}) \subseteq \mathbb{N}$.

Lemma: Decide non-negativity

An almost linear function is non-negative on $\mathbb{N}$ iff $e(E \cap \mathbb{N}) \subseteq \mathbb{N}$ and either $a = 0$, $b \geq 0$, or $a > 0$, $\{n \in \mathbb{N} \mid an + b < 0\} \subseteq E$.

Case analysis for deciding $w(\ell) > w(r)$

Theorem

For an alphabet Σ of size two, $a \in \Sigma$, a rewrite system R over Σ, and an almost linear function w, if w is non-negative on $\mathbb{N}$ and $w(\ell) > w(r)$ for each rule $\ell \to r$ in $\mathrm{slit}_a(R)$, then R is terminating, and R has linear derivational complexity.

Case analysis for deciding $w(\ell) > w(r)$

Theorem

For an alphabet Σ of size two, $a \in \Sigma$, a rewrite system R over Σ, and an almost linear function w, if w is non-negative on $\mathbb{N}$ and $w(\ell) > w(r)$ for each rule $\ell \to r$ in $\mathrm{slit}_a(R)$, then R is terminating, and R has linear derivational complexity.

Lemma: Closure properties

For an almost linear function w and a constant c, also $\lambda n.w(n) + c$ and $\lambda n.w(n + c)$ are almost linear.

Case analysis for deciding $w(\ell) > w(r)$

Theorem

For an alphabet Σ of size two, $a \in \Sigma$, a rewrite system R over Σ, and an almost linear function w, if w is non-negative on $\mathbb{N}$ and $w(\ell) > w(r)$ for each rule $\ell \to r$ in $\text{slit}_a(R)$, then R is terminating, and R has linear derivational complexity.

Example: Case analysis for Zantema_06/17 from TPDB

$w : 0 \mapsto 0, 1 \mapsto 1$, else $n \mapsto 4n - 5$ succeeds:
Among the 15 rules is $aba \to bbb$, so $\text{slit}_b(R)$ contains

$$\{(p+1, 1+q) \to (p, 0, 0, q) \mid p, q \in \mathbb{N}\}$$

Case analysis: p, q are 0 or 1 or > 1. E. g., for $p = q = 1$,

$$w((1+1, 1+1)) = w(2) + w(2) = 6 > 2$$
$$= w(1) + w(0) + w(0) + w(1) = w((1, 0, 0, 1))$$

More examples from TPDB

Bouchare_06/11: Large number of exceptions

split_a with
$w : 0 \mapsto 5, 1 \mapsto 4, 2 \mapsto 16, 3 \mapsto 17, 4 \mapsto 27, 5 \mapsto 30, 6 \mapsto 38,$
else $n \mapsto 6n + 1$

More examples from TPDB

Bouchare_06/11: Large number of exceptions

split_a with
$w : 0 \mapsto 5, 1 \mapsto 4, 2 \mapsto 16, 3 \mapsto 17, 4 \mapsto 27, 5 \mapsto 30, 6 \mapsto 38,$
$\text{else } n \mapsto 6n + 1$

ICFP_2010/136497: Even larger number of exceptions

split_1 with
$w : 0 \mapsto 0, 1 \mapsto 24, 2 \mapsto 42, 3 \mapsto 60, 4 \mapsto 76, 5 \mapsto 93,$
$6 \mapsto 109, 7 \mapsto 125, 8 \mapsto 142, \text{ else } n \mapsto 17n + 5$
(unsolved during termination competitions 2016/17/18)

More examples from TPDB

Bouchare_06/11: Large number of exceptions

split_a with
$w : 0 \mapsto 5, 1 \mapsto 4, 2 \mapsto 16, 3 \mapsto 17, 4 \mapsto 27, 5 \mapsto 30, 6 \mapsto 38,$
else $n \mapsto 6n + 1$

ICFP_2010/136497: Even larger number of exceptions

split_1 with
$w : 0 \mapsto 0, 1 \mapsto 24, 2 \mapsto 42, 3 \mapsto 60, 4 \mapsto 76, 5 \mapsto 93,$
$6 \mapsto 109, 7 \mapsto 125, 8 \mapsto 142,$ else $n \mapsto 17n + 5$
(unsolved during termination competitions 2016/17/18)

Bouchare_06/14: Trade-off ...

... between higher coefficient versus more exceptions:

- split_a with $w : 0 \mapsto 6$, else $n \mapsto 10n - 1$, or
- split_b with $w : 0 \mapsto 6, 3 \mapsto 13$, else $x \mapsto 3x + 2$

Prototype implementation within MultumNonMulta

- Experiments show rather limited applicability of the approach, but proofs are (mostly) found instantaneously.
- `SRS Standard` in TPDB 10.5 contains 358 systems over a two-letter alphabet: 22 can be solved by the protoype.

Prototype implementation within MultumNonMulta

- Experiments show rather limited applicability of the approach, but proofs are (mostly) found instantaneously.
- `SRS Standard` in TPDB 10.5 contains 358 systems over a two-letter alphabet: 22 can be solved by the protoype.

Algorithms for finding almost linear weight functions

Prototype implementation within MultumNonMulta

- Experiments show rather limited applicability of the approach, but proofs are (mostly) found instantaneously.
- `SRS Standard` in TPDB 10.5 contains 358 systems over a two-letter alphabet: 22 can be solved by the protoype.

Algorithms for finding almost linear weight functions

- Complete enumeration: Bounds for coefficient and constant of linear part, and domain and codomain of exception part.

Prototype implementation within MultumNonMulta

- Experiments show rather limited applicability of the approach, but proofs are (mostly) found instantaneously.
- `SRS Standard` in TPDB 10.5 contains 358 systems over a two-letter alphabet: 22 can be solved by the protoype.

Algorithms for finding almost linear weight functions

- Complete enumeration: Bounds for coefficient and constant of linear part, and domain and codomain of exception part.
- Integer constraint solving using the GNU Linear Programming Kit: Find the exception part by solving finite subproblems: $slit_a^n(\ell \to r) = \{p + \mathrm{slit}_a(\ell) + q \to p + \mathrm{slit}_a(r) + q \mid p, q \leq n\}$

Conclusion

- Termination proofs for non-simply terminating systems.
- Proof principle is almost self-explanatory.
- If successful: Simple proofs with short proof certificates.
- Limited applicability: Linear derivational complexity only.

Conclusion

- Termination proofs for non-simply terminating systems.
- Proof principle is almost self-explanatory.
- If successful: Simple proofs with short proof certificates.
- Limited applicability: Linear derivational complexity only.

Discussion and extensions

- Orthogonal to match-bounds (Geser/H/Waldmann'04).
- Related work: Waldmann/Zantema, Contejean et al.
- $R = \{abb \to a, aa \to bbb, bba \to aba\}$ (Bouchare_06/05):
  $w : 2n \mapsto n+3,\ 2n+1 \mapsto n$
- Relative termination / Ring rewriting
  (Zantema/Bruggink/Koenig'14) / ...

Conclusion

- Termination proofs for non-simply terminating systems.
- Proof principle is almost self-explanatory.
- If successful: Simple proofs with short proof certificates.
- Limited applicability: Linear derivational complexity only.

Discussion and extensions

- Orthogonal to match-bounds (Geser/H/Waldmann'04).
- Related work: Waldmann/Zantema, Contejean et al.
- $R = \{abb \to a, aa \to bbb, bba \to aba\}$ (`Bouchare_06/05`):
  $w : 2n \mapsto n + 3,\ 2n + 1 \mapsto n$
- Relative termination / Ring rewriting
  (Zantema/Bruggink/Koenig'14) / ...

- Embrace infinity.
- Don't –without need– restrict your theorems to the finite case.