Transforming Cycle Rewriting into String Rewriting

David Sabel[1] and Hans Zantema[2,3]

[1]Goethe University Frankfurt, Germany

[2]TU Eindhoven, The Netherlands
[3]Radboud University Nijmegen, The Netherlands

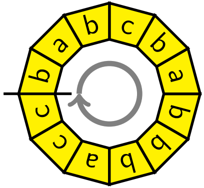
RTA 2015, Warsaw, Poland

Cycle Rewriting

A **cycle** is a string in which
the start and end are connected.

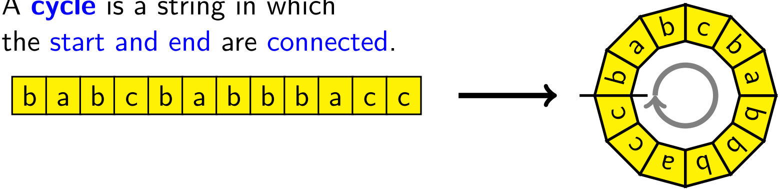
b	a	b	c	b	a	b	b	b	a	c	c

$\longrightarrow$



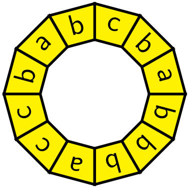
Cycle Rewriting

A **cycle** is a string in which the start and end are connected.

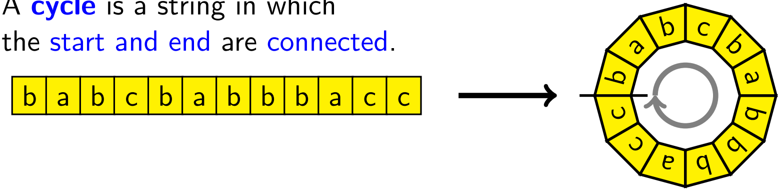


Cycle rewriting $\circ\!\!\rightarrow$

applies string rewrite rules to cycles, e.g. $R = \{\text{cba} \rightarrow \text{aabbcc}\}$

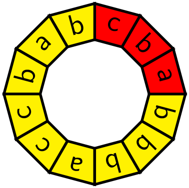


A **cycle** is a string in which the start and end are connected.



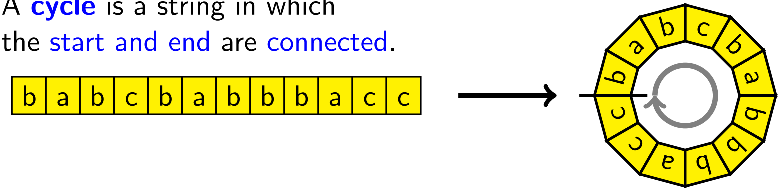
Cycle rewriting $\circ\!\!\rightarrow$

applies string rewrite rules to cycles, e.g. $R = \{\text{cba} \rightarrow \text{aabbcc}\}$

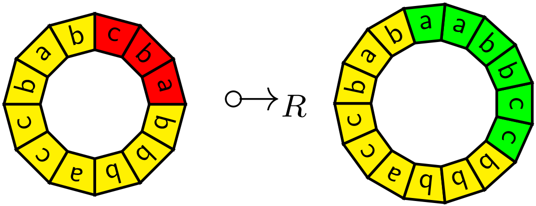


Cycle Rewriting

A **cycle** is a string in which
the start and end are connected.



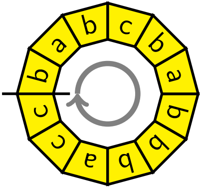
Cycle rewriting $\circ\!\!\rightarrow$
applies string rewrite rules to cycles, e.g. $R = \{\text{cba} \to \text{aabbcc}\}$



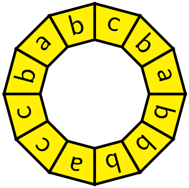
A **cycle** is a string in which
the start and end are connected.

b	a	b	c	b	a	b	b	b	a	c	c

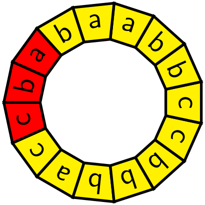
$\longrightarrow$



Cycle rewriting $\circ\!\!\rightarrow$
applies string rewrite rules to cycles, e.g. $R = \{\text{cba} \to \text{aabbcc}\}$

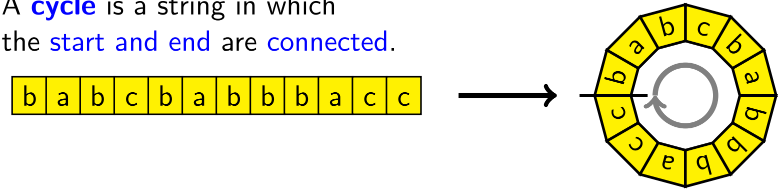


$\circ\!\!\longrightarrow_R$

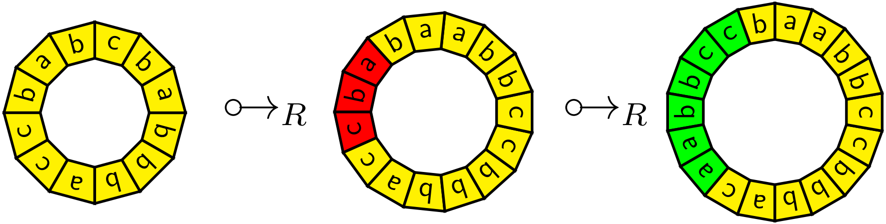


Cycle Rewriting

A **cycle** is a string in which
the start and end are connected.



Cycle rewriting $\circ\!\!\rightarrow$
applies string rewrite rules to cycles, e.g. $R = \{\text{cba} \to \text{aabbcc}\}$

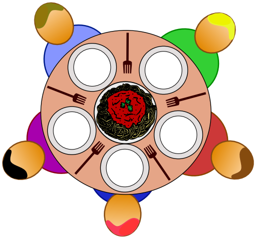


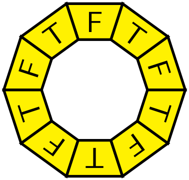
Applications of Cycle Rewriting

- Termination analysis for graph transformation systems
  [Bruggink,König,Zantema 2014, IFIP TCS]

Applications of Cycle Rewriting

- Termination analysis for graph transformation systems [Bruggink,König,Zantema 2014, IFIP TCS]
- Some systems are naturally cycle rewrite systems:



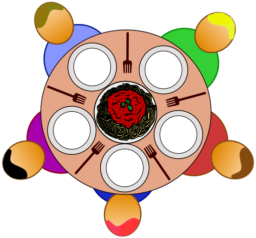


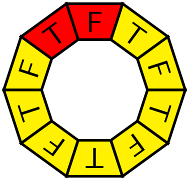
T: thinking philosopher
F: fork
L: philosopher has left fork
E: eating philosopher

Rewrite rules:

T F	$\to$	L	(pick up left fork)
F L	$\to$	E	(pick up right fork and eat)
E	$\to$	F T F	(stop eating and put down forks)

- Termination analysis for graph transformation systems [Bruggink,König,Zantema 2014, IFIP TCS]
- Some systems are naturally cycle rewrite systems:



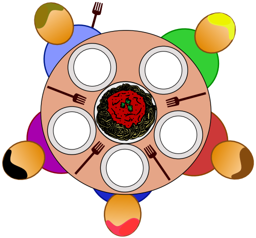


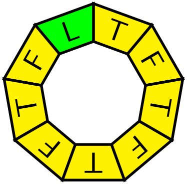
T: thinking philosopher
F: fork
L: philosopher has left fork
E: eating philosopher

Rewrite rules:

T F	$\rightarrow$	L	(pick up left fork)
F L	$\rightarrow$	E	(pick up right fork and eat)
E	$\rightarrow$	F T F	(stop eating and put down forks)

- Termination analysis for graph transformation systems [Bruggink,König,Zantema 2014, IFIP TCS]
- Some systems are naturally cycle rewrite systems:



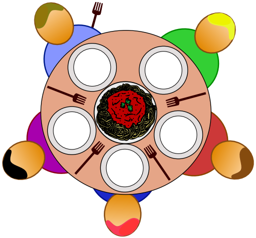


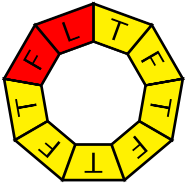
T: thinking philosopher
F: fork
L: philosopher has left fork
E: eating philosopher

Rewrite rules:

T F	$\rightarrow$	L	(pick up left fork)
F L	$\rightarrow$	E	(pick up right fork and eat)
E	$\rightarrow$	F T F	(stop eating and put down forks)

- Termination analysis for graph transformation systems [Bruggink,König,Zantema 2014, IFIP TCS]
- Some systems are naturally cycle rewrite systems:





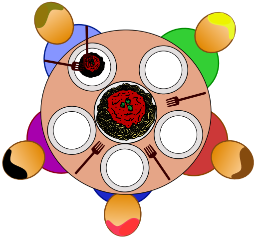
T: thinking philosopher
F: fork
L: philosopher has left fork
E: eating philosopher

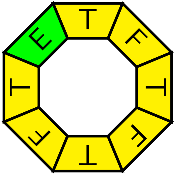
Rewrite rules:

T F	$\to$	L	(pick up left fork)
F L	$\to$	E	(pick up right fork and eat)
E	$\to$	F T F	(stop eating and put down forks)

Applications of Cycle Rewriting

- Termination analysis for graph transformation systems [Bruggink,König,Zantema 2014, IFIP TCS]
- Some systems are naturally cycle rewrite systems:



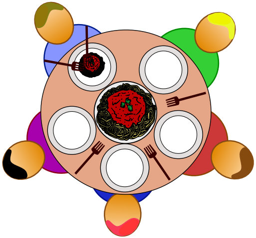


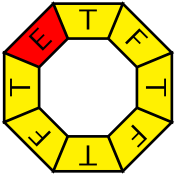
T: thinking philosopher
F: fork
L: philosopher has left fork
E: eating philosopher

Rewrite rules:

T F	$\rightarrow$	L	(pick up left fork)
F L	$\rightarrow$	E	(pick up right fork and eat)
E	$\rightarrow$	F T F	(stop eating and put down forks)

- Termination analysis for graph transformation systems
  [Bruggink,König,Zantema 2014, IFIP TCS]
- Some systems are naturally cycle rewrite systems:





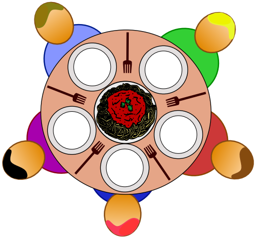
T: thinking philosopher
F: fork
L: philosopher has left fork
E: eating philosopher

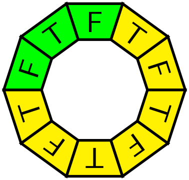
Rewrite rules:

T F	$\to$	L	(pick up left fork)
F L	$\to$	E	(pick up right fork and eat)
E	$\to$	F T F	(stop eating and put down forks)

Applications of Cycle Rewriting

- Termination analysis for graph transformation systems [Bruggink,König,Zantema 2014, IFIP TCS]
- Some systems are naturally cycle rewrite systems:



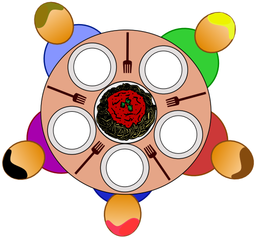


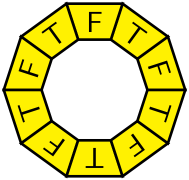
T: thinking philosopher
F: fork
L: philosopher has left fork
E: eating philosopher

Rewrite rules:

T F	$\to$	L	(pick up left fork)
F L	$\to$	E	(pick up right fork and eat)
E	$\to$	F T F	(stop eating and put down forks)

- Termination analysis for graph transformation systems [Bruggink,König,Zantema 2014, IFIP TCS]
- Some systems are naturally cycle rewrite systems:





T: thinking philosopher
F: fork
L: philosopher has left fork
E: eating philosopher

Rewrite rules:

T F	$\to$	L	(pick up left fork)
F L	$\to$	E	(pick up right fork and eat)
E	$\to$	F T F	(stop eating and put down forks)

Let Σ be an alphabet, R be an SRS over Σ

- $\boldsymbol{u \sim v}$ = strings $u, v \in \Sigma^*$ represent the same cycle:

  $$u \sim v \text{ iff } \exists w_1, w_2 : u = w_1 w_2 \text{ and } v = w_2 w_1$$

- **cycle** $[\boldsymbol{u}]$ = equivalence class of string u w.r.t. $\sim$
- **cycle rewrite relation** $\circ\!\!\to_R \subseteq (\Sigma/\!\sim \times\, \Sigma/\!\sim)$ of R:

$$[u] \circ\!\!\to_R [v] \text{ iff } \exists w \in \Sigma^* : u \sim \ell w, (\ell \to r) \in R, \text{ and } rw \sim v$$

- $\circ\!\!\longrightarrow_R$ is **non-terminating** iff there exists an infinite sequence

$$[u_0] \circ\!\!\longrightarrow_R [u_1] \circ\!\!\longrightarrow_R [u_2] \circ\!\!\longrightarrow_R \cdots$$

- Otherwise, $\circ\!\!\longrightarrow_R$ is **terminating**.

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

- $\circ\!\!\rightarrow_R$ is **non-terminating** iff there exists an infinite sequence

$$[u_0] \circ\!\!\rightarrow_R [u_1] \circ\!\!\rightarrow_R [u_2] \circ\!\!\rightarrow_R \cdots$$

- Otherwise, $\circ\!\!\rightarrow_R$ is **terminating**.
- Cycle-termination **is different** from string-termination:

  for $R = \{ab \to ba\}$

  - $\to_R$ is terminating, but
  - $\circ\!\!\rightarrow_R$ is non-terminating
- But non-termination of $\to_R$ implies non-termination of $\circ\!\!\rightarrow_R$

Termination techniques

- arctic and tropical matrix interpretations based on type-graphs
- implemented in torpacyc, iteratively removes rewrite rules using relative termination
- technique can only remove rules which are applied at most polynomially often in any derivation

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Termination techniques

- arctic and tropical matrix interpretations based on type-graphs
- implemented in torpacyc, iteratively removes rewrite rules using relative termination
- technique can only remove rules which are applied at most polynomially often in any derivation

Complexity

Transformation ϕ on SRSs R ("string rewriting $\to$ cycle rewriting") s.t.

$$\to_R \text{ is string-terminating } \iff \circ\!\!\to_{\phi(R)} \text{ is cycle-terminating}$$

Consequences:

- proving cycle-termination is **at least as hard as string-termination**
- proving cycle-termination is **undecidable**

Transformational approach

1. reduce cycle-termination to string-termination
2. apply state-of-the-art ATPs to prove string-termination

required: transformation ψ : "cycle rewriting $\to$ string rewriting" which is

sound: $\to_{\psi(R)}$ is string-terminating $\implies$ $\circ\!\!\to_R$ is cycle-terminating

complete: $\circ\!\!\to_R$ is cycle-terminating $\implies$ $\to_{\psi(R)}$ is string-terminating

We provide three sound and complete transformations split, rotate, shift

Transformational approach

1. reduce cycle-termination to string-termination
2. apply state-of-the-art ATPs to prove string-termination

required: transformation ψ : "cycle rewriting $\to$ string rewriting" which is

sound: $\to_{\psi(R)}$ is string-terminating $\implies$ $\circ\!\!\to_R$ is cycle-terminating

complete: $\circ\!\!\to_R$ is cycle-terminating $\implies$ $\to_{\psi(R)}$ is string-terminating

We provide three sound and complete transformations split, rotate, shift

Trace-decreasing matrix interpreations

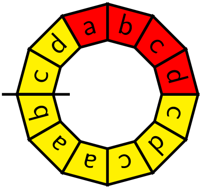
- following a suggestion of Johannes Waldmann
- extend the matrix interpretations from [Zantema,König,Bruggink 2014,RTA]

The Transformation Split: Idea

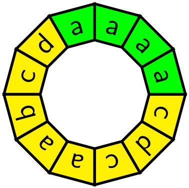
For a cycle rewrite step $[u_1] \circ\!\!\rightarrow_{\{\ell \to r\}} [u_2]$ and $v_1 \in [u_1]$

For a cycle rewrite step $[u_1] \circ\!\!\to_{\{\ell\to r\}} [u_2]$ and $v_1 \in [u_1]$

- **case 1:** $v_1 \to_{\{\ell\to r\}} v_2$ where $v_2 \in [u_2]$



$\circ\!\!\to_{\{\mathsf{abcd}\to\mathsf{aaaa}\}}$



c	d	a	b	c	d	c	d	c	a	a	b

$\to_{\{\mathsf{abcd}\to\mathsf{aaaa}\}}$

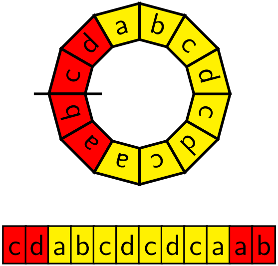
c	d	a	a	a	a	c	d	c	a	a	b

The Transformation Split: Idea

For a cycle rewrite step $[u_1] \circlearrowright\!\!\to_{\{\ell \to r\}} [u_2]$ and $v_1 \in [u_1]$

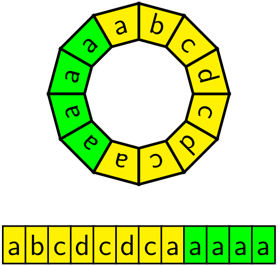
- **case 2:** we can split $\ell = \ell_A \ell_B$ s.t.

$$v_1 = \ell_B u \ell_A \to_{\{\ell_B \to \varepsilon\}} u \ell_A \to_{\{\ell_A \to r\}} ur \text{ where } ur \in [u_2]$$



$\circ\!\!\to_{\{\text{abcd} \to \text{aaaa}\}}$

ab$\to$aaaa cd$\to\varepsilon$

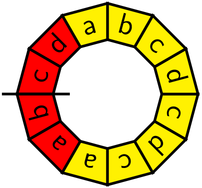


cdabcdcdcaab $\to_{\{\text{cd} \to \varepsilon\}}$ abcdcdcaab $\to_{\{\text{ab} \to \text{aaaa}\}}$ abcdcdaaaa

For a cycle rewrite step $[u_1] \circ\!\!\rightarrow_{\{\ell\to r\}} [u_2]$ and $v_1 \in [u_1]$

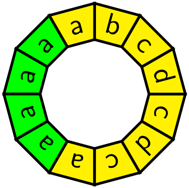
- **case 2:** we can split $\ell = \ell_A \ell_B$ s.t.

$$v_1 = \overbrace{\ell_B u \ell_A \rightarrow_{\{\ell_B\to\varepsilon\}} u\ell_A}^{\text{prefix string rewrite step}} \underbrace{\rightarrow_{\{\ell_A\to r\}} ur}_{\text{suffix string rewrite step}} \text{ where } ur \in [u_2]$$



$\circ\!\!\rightarrow$ {abcd→aaaa}

ab→aaaa cd→ε



c	d	a	b	c	d	c	d	c	a	a	b

a	b	c	d	c	d	c	a	a	a	a	a

cdabcdcdcaab $\rightarrow_{\{\text{cd}\to\varepsilon\}}$ abcdcdcaab $\rightarrow_{\{\text{ab}\to\text{aaaa}\}}$ abcdcdaaaa

Naive (but sound) transformation:

- Add all rewrite rules $(\ell \to r)$
- Add all splitting rules $(\ell_A \to r)$ and $(\ell_B \to \varepsilon)$ for $\ell = \ell_A \ell_B$

$\Rightarrow$ results in **non-terminating SRSs** in most of the cases
(i.e. whenever r contains some prefix ℓ_A)

Naive (but sound) transformation:

- Add all rewrite rules $(\ell \to r)$
- Add all splitting rules $(\ell_A \to r)$ and $(\ell_B \to \varepsilon)$ for $\ell = \ell_A \ell_B$

$\Rightarrow$ results in **non-terminating SRSs** in most of the cases
(i.e. whenever r contains some prefix ℓ_A)

Requirements for a better transformation (and for completeness)

- ensure that split rules are **only** applied to a **prefix** or a **suffix**, resp.
  $\Rightarrow$ surround the string by fresh begin symbol B and end symbol E
- **synchronize** the application of the prefix and the suffix rewrite step
  $\Rightarrow$ use fresh symbols $\overline{a}$ for $a \in \Sigma$, and W, L and $\mathsf{R}_{i,j}$

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Definition of the transformation split(.)

For an SRS R over alphabet Σ, the SRS $\mathsf{split}(R)$ over $\Sigma_{\mathsf{split}} = \Sigma \cup \overline{\Sigma} \cup \{\mathsf{B}, \mathsf{E}, \mathsf{L}, \mathsf{W}, \mathsf{R}_{i,j}\}$ is constructed as follows:

- Let $(\ell \to r) \in R$ be the i^{th} rule of R:
  - add rule $\ell \to r$ (for case 1)
  - for every splitting $\ell = \ell_A \ell_B$ with $|\ell_A| = j$, add the rules:

$$\begin{array}{lll} \mathsf{B}\ell_B & \to \mathsf{W}\mathsf{R}_{i,j} & \text{(prefix rewrite step)} \\ \mathsf{R}_{i,j}\,a & \to \overline{a}\mathsf{R}_{i,j} & \text{(synchronize, shift } \mathsf{R}_{i,j} \text{ in front of the suffix)} \\ \mathsf{R}_{i,j}\,\ell_A\,\mathsf{E} & \to \mathsf{L}\,r\,\mathsf{E} & \text{(suffix rewrite step)} \end{array}$$

- add rules $\overline{a}\mathsf{L} \to \mathsf{L}a$ for all $a \in \Sigma$ (clean up)
- add rule $\mathsf{WL} \to \mathsf{B}$ (finish)

Theorem

The transformation split **is sound and complete**,
i.e. $\to_{\mathsf{split}(R)}$ is string-terminating iff $\circ\!\!\to_R$ is cycle-terminating.

- Soundness follows by construction:

$$[u] \circ\!\!\to_R [v] \implies \mathsf{B}\,u\,\mathsf{E} \to^{+}_{\mathsf{split}(R)} \mathsf{B}\,v'\,\mathsf{E} \text{ where } v' \sim v$$

- Completeness can be shown by
  - **type introduction** [Zantema 1994, JSC]
  - a mapping $\Phi :: \Sigma^*_{\mathsf{split}} \to \Sigma^*$ with

$$\forall u :: T : \quad u \to_{\mathsf{split}(R)} u' \quad \implies \quad [\Phi(u)] \circ\!\!\to^{*}_{R} [\Phi(u')]$$

Trace-Decreasing Matrix Interpretations

- $M_d :=$ all $d \times d$ matrices A over $\mathbb{N}$ s.t. $A_{11} > 0$
- for $A, B \in M_d$,

$$\begin{array}{lcl} A > B & \iff & A_{11} > B_{11} \wedge \forall i,j : A_{ij} \geq B_{ij} \\ A \geq B & \iff & \forall i,j : A_{ij} \geq B_{ij} \end{array}$$

- $M_d :=$ all $d \times d$ matrices A over $\mathbb{N}$ s.t. $A_{11} > 0$
- for $A, B \in M_d$,

$$\begin{array}{lcl} A > B & \iff & A_{11} > B_{11} \land \forall i,j : A_{ij} \geq B_{ij} \\ A \geq B & \iff & \forall i,j : A_{ij} \geq B_{ij} \end{array}$$

- a **matrix interpretation** $\langle\cdot\rangle : \Sigma \to M_d$ is extended to strings as

$$\langle\varepsilon\rangle = I \quad \text{and} \quad \langle ua\rangle = \langle u\rangle \times \langle a\rangle \text{ for all } u \in \Sigma^*,\ a \in \Sigma$$

where I is the identity matrix, $\times$ is matrix multiplication

Theorem

Let $R' \subseteq R$ be SRSs over Σ and let $\langle\cdot\rangle : \Sigma \to M_d$ such that

- $\circ\!\!\to_{R'}$ is terminating,
- $\langle\ell\rangle \geq \langle r\rangle$ for all $(\ell \to r) \in R'$, and
- $\langle\ell\rangle > \langle r\rangle$ for all $(\ell \to r) \in R \setminus R'$.

Then $\circ\!\!\to_{R}$ is terminating.

Theorem

Let $R' \subseteq R$ be SRSs over Σ and let $\langle \cdot \rangle : \Sigma \to M_d$ such that

- $\circ\!\!\to_{R'}$ is terminating,
- $\langle \ell \rangle \geq \langle r \rangle$ for all $(\ell \to r) \in R'$, and
- $\langle \ell \rangle > \langle r \rangle$ for all $(\ell \to r) \in R \setminus R'$.

Then $\circ\!\!\to_R$ is terminating.

Proof: The main observations are

- $\mathsf{trace}(\langle a \rangle \times \langle u \rangle) = \mathsf{trace}(\langle u \rangle \times \langle a \rangle)$ and thus $\mathsf{trace}(\langle u \rangle) = \mathsf{trace}(\langle v \rangle)$ if $u \sim v$
- $>, \geq$ are stable w.r.t $\times$, and thus $\langle \ell \rangle > \langle r \rangle \implies \langle \ell w \rangle > \langle r w \rangle$
- $[u] \circ\!\!\to_{R'} [v] \implies \mathsf{trace}(\langle u \rangle) \geq \mathsf{trace}(\langle v \rangle)$, and
- $[u] \circ\!\!\to_{R \setminus R'} [v] \implies \mathsf{trace}(\langle u \rangle) > \mathsf{trace}(\langle v \rangle)$

Improvements and Limitations

Trace-decreasing matrix interpretations

- **can remove rules which are applied exponentially often** (improves [Zantema, König, Bruggink 2014,RTA])
- **impossible** to remove rules which are applied more often

Trace-decreasing matrix interpretations

- **can remove rules which are applied exponentially often** (improves [Zantema, König, Bruggink 2014,RTA])
- **impossible** to remove rules which are applied more often

Example (adapted from [Hofbauer and Waldmann 2006,RTA])

$$\begin{aligned}\mathcal{R} := \;& \phi(\{ab \to bca, cb \to bbc\}) \\ = \;& \{RE \to LE, aL \to La', bL \to Lb', cL \to Lc', Ra' \to aR, \\ & Rb' \to bR, Rc' \to cR, abL \to bcaR, cbL \to bbcR\}\end{aligned}$$

- has cycle rewrite derivations where the number of rule applications is a **tower of exponentials for each rule**
- **impossible** to prove cycle termination by trace-decreasing matrix interpretations

Trace-decreasing matrix interpretations

- **can remove rules which are applied exponentially often** (improves [Zantema, König, Bruggink 2014,RTA])
- **impossible** to remove rules which are applied more often

Example (adapted from [Hofbauer and Waldmann 2006,RTA])

$$\begin{aligned}\mathcal{R} := {}& \phi(\{ab \to bca, cb \to bbc\}) \\ = {}& \{RE \to LE, aL \to La', bL \to Lb', cL \to Lc', Ra' \to aR, \\ & Rb' \to bR, Rc' \to cR, abL \to bcaR, cbL \to bbcR\}\end{aligned}$$

- has cycle rewrite derivations where the number of rule applications is a **tower of exponentials for each rule**
- **impossible** to prove cycle termination by trace-decreasing matrix interpretations
- **but** AProVE **proves** string **termination of** split($\mathcal{R}$)
  $\Rightarrow$ **transformational approach succeeds**

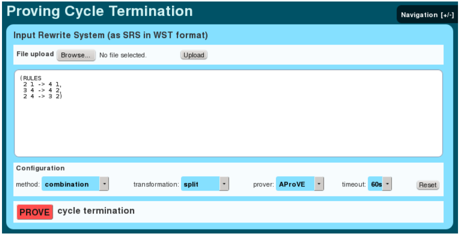
GOETHE UNIVERSITÄT FRANKFURT AM MAIN

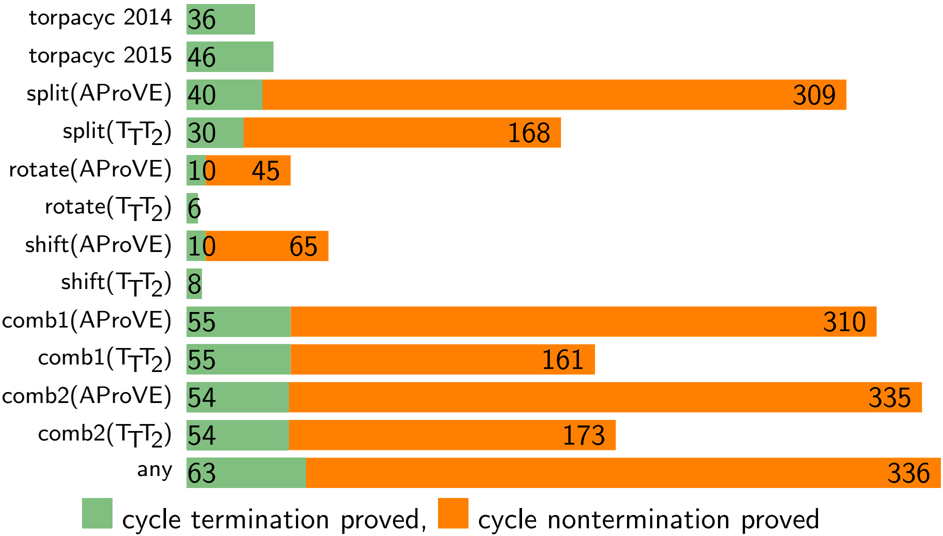
Techniques

- torpacyc: trace decreasing matrix interpretations
- transformations split, rotate, shift with AProVE and $\mathrm{T_TT_2}$
- combination 1: first torpacyc then transformation split
- combination 2: like combination 1, but first string-nontermination check by AProVE, or $\mathrm{T_TT_2}$, resp.

Tools and webinterface available via

http://www.ki.cs.uni-frankfurt.de/research/cycsrs





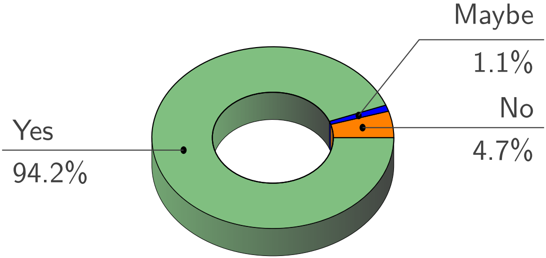
1315 problems, timeout 60sec, 916 problems remain open

50000 **randomly generated** problems

- of size 12 with $|\Sigma| = 3$
- no obviously nonterminating problems

Results

- rotate and shift show termination of **74 %** of the problems
- torpacyc and split show termination of **94 %** of the problems
- In **total** (combining all results):



GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Conclusion

- new techniques to **prove cycle termination**
- three **sound and complete transformations** from cycle into string rewriting
- transformation split seems to be **useful in practice**
- **trace-decreasing matrix interpretations**
- new techniques **solve problems** for which the earlier techniques failed

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Conclusion

- new techniques to **prove cycle termination**
- three **sound and complete transformations** from cycle into string rewriting
- transformation split seems to be **useful in practice**
- **trace-decreasing matrix interpretations**
- new techniques **solve problems** for which the earlier techniques failed

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Conclusion

- new techniques to **prove cycle termination**
- three **sound and complete transformations** from cycle into string rewriting
- transformation split seems to be **useful in practice**
- **trace-decreasing matrix interpretations**
- new techniques **solve problems** for which the earlier techniques failed

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Conclusion

- new techniques to **prove cycle termination**
- three **sound and complete transformations** from cycle into string rewriting
- transformation split seems to be **useful in practice**
- **trace-decreasing matrix interpretations**
- new techniques **solve problems** for which the earlier techniques failed

Conclusion & Future Work

Conclusion

- new techniques to **prove cycle termination**
- three **sound and complete transformations** from cycle into string rewriting
- transformation split seems to be **useful in practice**
- **trace-decreasing matrix interpretations**
- new techniques **solve problems** for which the earlier techniques failed

Future work

- extend the benchmark problem set
- specific methods for cycle non-termination
- applications for cycle rewriting