Sparse Tiling Through Overlap Closures for Termination of String Rewriting

Alfons Geser (HTWK Leipzig),
Dieter Hofbauer (ASW BA Saarland),
Johannes Waldmann (HTWK Leipzig)

FSCD 2019

Preliminaries: Termination

- relation $\to$ is terminating (strongly normalizing)
  := there are no infinite $\to$-chains
  notations: $\mathrm{SN}(\to), \mathrm{SN}(\to_R), \mathrm{SN}(R)$.
- methods for proving termination of rewriting:
  - syntactical (precedence on symbols)
  - semantical (interprete symbols by functions over well-founded domain)
  - transformational ($\mathrm{SN}(R) \Leftarrow \mathrm{SN}(R')$)
- in particular: transformation that increases signature, give more room to pick predecence or interpretation
- …by *tiling*: new signature consists of *tiles* (blocks of adjacent letters)

Preliminaries: Termination

- relation $\to$ is terminating (strongly normalizing)
  := there are no infinite $\to$-chains
  notations: $\mathrm{SN}(\to), \mathrm{SN}(\to_R), \mathrm{SN}(R)$.
- methods for proving termination of rewriting:
  - syntactical (precedence on symbols)
  - semantical (interprete symbols by functions over well-founded domain)
  - transformational ($\mathrm{SN}(R) \Leftarrow \mathrm{SN}(R')$)
- in particular: transformation that increases signature, give more room to pick predecence or interpretation
- ...by *tiling*: new signature consists of *tiles* (blocks of adjacent letters)

Preliminaries: Termination

- relation $\to$ is terminating (strongly normalizing)
  := there are no infinite $\to$-chains
  notations: $\mathrm{SN}(\to), \mathrm{SN}(\to_R), \mathrm{SN}(R)$.
- methods for proving termination of rewriting:
  - syntactical (precedence on symbols)
  - semantical (interprete symbols by functions over well-founded domain)
  - transformational ($\mathrm{SN}(R) \Leftarrow \mathrm{SN}(R')$)
- in particular: transformation that increases signature, give more room to pick predecence or interpretation
- …by *tiling*: new signature consists of *tiles* (blocks of adjacent letters)

Preliminaries: Termination

- relation $\to$ is terminating (strongly normalizing)
  := there are no infinite $\to$-chains
  notations: $\mathrm{SN}(\to), \mathrm{SN}(\to_R), \mathrm{SN}(R)$.
- methods for proving termination of rewriting:
  - syntactical (precedence on symbols)
  - semantical (interprete symbols by functions over well-founded domain)
  - transformational ($\mathrm{SN}(R) \Leftarrow \mathrm{SN}(R')$)
- in particular: transformation that increases signature, give more room to pick predecence or interpretation
- …by *tiling*: new signature consists of *tiles* (blocks of adjacent letters)

Preliminaries: Termination

- relation $\to$ is terminating (strongly normalizing)
  := there are no infinite $\to$-chains
  notations: $\mathrm{SN}(\to), \mathrm{SN}(\to_R), \mathrm{SN}(R)$.
- methods for proving termination of rewriting:
  - syntactical (precedence on symbols)
  - semantical (interprete symbols by functions over well-founded domain)
  - transformational ($\mathrm{SN}(R) \Leftarrow \mathrm{SN}(R')$)
- in particular: transformation that increases signature, give more room to pick predecence or interpretation
- ... by *tiling*: new signature consists of *tiles* (blocks of adjacent letters)

Preliminaries: Termination

- relation $\to$ is terminating (strongly normalizing)
  := there are no infinite $\to$-chains
  notations: $\mathrm{SN}(\to), \mathrm{SN}(\to_R), \mathrm{SN}(R)$.
- methods for proving termination of rewriting:
  - syntactical (precedence on symbols)
  - semantical (interprete symbols by functions over well-founded domain)
  - transformational ($\mathrm{SN}(R) \Leftarrow \mathrm{SN}(R')$)
- in particular: transformation that increases signature, give more room to pick predecence or interpretation
- ... by *tiling*: new signature consists of *tiles* (blocks of adjacent letters)

Preliminaries: Termination

- relation $\to$ is terminating (strongly normalizing)
  := there are no infinite $\to$-chains
  notations: $\mathrm{SN}(\to), \mathrm{SN}(\to_R), \mathrm{SN}(R)$.
- methods for proving termination of rewriting:
  - syntactical (precedence on symbols)
  - semantical (interprete symbols by functions over well-founded domain)
  - transformational ($\mathrm{SN}(R) \Leftarrow \mathrm{SN}(R')$)
- in particular: transformation that increases signature, give more room to pick predecence or interpretation
- ...by *tiling*: new signature consists of *tiles* (blocks of adjacent letters)

Preliminaries: Tiling

- $S = \{aa \to aba\}$ does not remove letters
- use tiles of width 2 (pairs of adjacent letters)
  $S_2 = \{[aa] \to [ab, ba]\}$, can simulate S-derivations
  S_2 removes letter $[aa]$: is terminating!
- in general: need (left and right) padding
  ex. from rule $ab \to ba$, create
  $[aa][ab][ba] \to [ab][ba][aa]$, $[aa][ab][bb] \to [ab][ba][ab]$,
  $[ba][ab][ba] \to [bb][ba][aa]$, $[ba][ab][bb] \to [bb][ba][ab]$
- instance: root labelling (Sternagel, Middeldorp, RTA 2008)
- our contribution:
  - use smaller set of tiles (for rewriting and for padding)
  - only those that appear in (certain) infinite derivations

Preliminaries: Tiling

- $S = \{aa \to aba\}$ does not remove letters
- use tiles of width 2 (pairs of adjacent letters)
  $S_2 = \{[aa] \to [ab, ba]\}$, can simulate S-derivations
  S_2 removes letter $[aa]$: is terminating!
- in general: need (left and right) padding
  ex. from rule $ab \to ba$, create
  $[aa][ab][ba] \to [ab][ba][aa], [aa][ab][bb] \to [ab][ba][ab]$,
  $[ba][ab][ba] \to [bb][ba][aa], [ba][ab][bb] \to [bb][ba][ab]$
- instance: root labelling (Sternagel, Middeldorp, RTA 2008)
- our contribution:
  - use smaller set of tiles (for rewriting and for padding)
  - only those that appear in (certain) infinite derivations

Preliminaries: Tiling

- $S = \{aa \to aba\}$ does not remove letters
- use tiles of width 2 (pairs of adjacent letters)
  $S_2 = \{[aa] \to [ab, ba]\}$, can simulate S-derivations
  S_2 removes letter $[aa]$: is terminating!
- in general: need (left and right) padding
  ex. from rule $ab \to ba$, create
  $[aa][ab][ba] \to [ab][ba][aa], [aa][ab][bb] \to [ab][ba][ab],$
  $[ba][ab][ba] \to [bb][ba][aa], [ba][ab][bb] \to [bb][ba][ab]$
- instance: root labelling (Sternagel, Middeldorp, RTA 2008)
- our contribution:
  - use smaller set of tiles (for rewriting and for padding)
  - only those that appear in (certain) infinite derivations

Preliminaries: Tiling

- $S = \{aa \to aba\}$ does not remove letters
- use tiles of width 2 (pairs of adjacent letters)
  $S_2 = \{[aa] \to [ab, ba]\}$, can simulate S-derivations
  S_2 removes letter $[aa]$: is terminating!
- in general: need (left and right) padding
  ex. from rule $ab \to ba$, create
  $[aa][ab][ba] \to [ab][ba][aa], [aa][ab][bb] \to [ab][ba][ab],$
  $[ba][ab][ba] \to [bb][ba][aa], [ba][ab][bb] \to [bb][ba][ab]$
- instance: root labelling (Sternagel, Middeldorp, RTA 2008)
- our contribution:
  - use smaller set of tiles (for rewriting and for padding)
  - only those that appear in (certain) infinite derivations

Preliminaries: Tiling

- $S = \{aa \to aba\}$ does not remove letters
- use tiles of width 2 (pairs of adjacent letters)
  $S_2 = \{[aa] \to [ab, ba]\}$, can simulate S-derivations
  S_2 removes letter $[aa]$: is terminating!
- in general: need (left and right) padding
  ex. from rule $ab \to ba$, create
  $[aa][ab][ba] \to [ab][ba][aa], [aa][ab][bb] \to [ab][ba][ab]$,
  $[ba][ab][ba] \to [bb][ba][aa], [ba][ab][bb] \to [bb][ba][ab]$
- instance: root labelling (Sternagel, Middeldorp, RTA 2008)
- our contribution:
  - use smaller set of tiles (for rewriting and for padding)
  - only those that appear in (certain) infinite derivations

Preliminaries: Tiling

- $S = \{aa \to aba\}$ does not remove letters
- use tiles of width 2 (pairs of adjacent letters)
  $S_2 = \{[aa] \to [ab, ba]\}$, can simulate S-derivations
  S_2 removes letter $[aa]$: is terminating!
- in general: need (left and right) padding
  ex. from rule $ab \to ba$, create
  $[aa][ab][ba] \to [ab][ba][aa], [aa][ab][bb] \to [ab][ba][ab],$
  $[ba][ab][ba] \to [bb][ba][aa], [ba][ab][bb] \to [bb][ba][ab]$
- instance: root labelling (Sternagel, Middeldorp, RTA 2008)
- our contribution:
  - use smaller set of tiles (for rewriting and for padding)
  - only those that appear in (certain) infinite derivations

Preliminaries: Tiling

- $S = \{aa \to aba\}$ does not remove letters
- use tiles of width 2 (pairs of adjacent letters)
  $S_2 = \{[aa] \to [ab, ba]\}$, can simulate S-derivations
  S_2 removes letter $[aa]$: is terminating!
- in general: need (left and right) padding
  ex. from rule $ab \to ba$, create
  $[aa][ab][ba] \to [ab][ba][aa], [aa][ab][bb] \to [ab][ba][ab],$
  $[ba][ab][ba] \to [bb][ba][aa], [ba][ab][bb] \to [bb][ba][ab]$
- instance: root labelling (Sternagel, Middeldorp, RTA 2008)
- our contribution:
  - use smaller set of tiles (for rewriting and for padding)
  - only those that appear in (certain) infinite derivations

Sparse Tiling: Definition and Motivation

- Ex. the bordered 3-tiles of string $w = bbaab$ are $\mathrm{btiles}_3(w) = \{\triangleleft\triangleleft b, \triangleleft bb, bba, aab, ab\triangleright, b\triangleright\triangleright\}$
- Def. [Zalcstein 1972] strictly locally testable language $\mathrm{Lang}(T) = \{w \mid \mathrm{btiles}(w) \subseteq T\}$
- this paper:
  - use such languages to over-approximate $R^*(L)$
  - represent T by finite automaton A,
  - … constructed by completion
  - semantically label R by the partial algebra of A
  - … to transform the termination problem of R on L
  - *sparse*: T is the set of tiles that occur in rhs of forward closures (overlap closures, resp.)
- application: Matchbox wins Termcomp 2019 for SRS

Sparse Tiling: Definition and Motivation

- Ex. the bordered 3-tiles of string $w = bbaab$ are $\mathrm{btiles}_3(w) = \{\triangleleft\triangleleft b, \triangleleft bb, bba, aab, ab\triangleright, b\triangleright\triangleright\}$
- Def. [Zalcstein 1972] strictly locally testable language $\mathrm{Lang}(T) = \{w \mid \mathrm{btiles}(w) \subseteq T\}$
- this paper:
  - use such languages to over-approximate $R^*(L)$
  - represent T by finite automaton A,
  - … constructed by completion
  - semantically label R by the partial algebra of A
  - … to transform the termination problem of R on L
  - sparse: T is the set of tiles that occur in the $\ldots$ of forward closures (overlap closures, resp.)
- application: Matchbox wins Termcomp 2019 for SRS

Sparse Tiling: Definition and Motivation

- Ex. the bordered 3-tiles of string $w = bbaab$ are $\text{btiles}_3(w) = \{\lhd\lhd b, \lhd bb, bba, aab, ab\rhd, b\rhd\rhd\}$
- Def. [Zalcstein 1972] strictly locally testable language $\text{Lang}(T) = \{w \mid \text{btiles}(w) \subseteq T\}$
- this paper:
  - use such languages to over-approximate $R^*(L)$
  - represent T by finite automaton A,
  - ... constructed by completion
  - semantically label R by the partial algebra of A
  - ... to transform the termination problem of R on L.
  - *sparse*: T is the set of tiles that occur in rhs of forward closures (overlap closures, resp.)
- application: Matchbox wins Termcomp 2019 for SRS

Sparse Tiling: Definition and Motivation

- Ex. the bordered 3-tiles of string $w = bbaab$ are $\mathrm{btiles}_3(w) = \{\triangleleft\triangleleft b, \triangleleft bb, bba, aab, ab\triangleright, b\triangleright\triangleright\}$
- Def. [Zalcstein 1972] strictly locally testable language $\mathrm{Lang}(T) = \{w \mid \mathrm{btiles}(w) \subseteq T\}$
- this paper:
  - use such languages to over-approximate $R^*(L)$
  - represent T by finite automaton A,
  - ... constructed by completion
  - semantically label R by the partial algebra of A
  - ... to transform the termination problem of R on L.
  - *sparse*: T is the set of tiles that occur in rhs of forward closures (overlap closures, resp.)
- application: Matchbox wins Termcomp 2019 for SRS

Sparse Tiling: Definition and Motivation

- Ex. the bordered 3-tiles of string $w = bbaab$ are $\text{btiles}_3(w) = \{\triangleleft\triangleleft b, \triangleleft bb, bba, aab, ab\triangleright, b\triangleright\triangleright\}$
- Def. [Zalcstein 1972] strictly locally testable language $\text{Lang}(T) = \{w \mid \text{btiles}(w) \subseteq T\}$
- this paper:
  - use such languages to over-approximate $R^*(L)$
  - represent T by finite automaton A,
  - ... constructed by completion
  - semantically label R by the partial algebra of A
  - ... to transform the termination problem of R on L.
  - *sparse*: T is the set of tiles that occur in rhs of forward closures (overlap closures, resp.)
- application: Matchbox wins Termcomp 2019 for SRS

Sparse Tiling: Definition and Motivation

- Ex. the bordered 3-tiles of string $w = bbaab$ are $\text{btiles}_3(w) = \{\triangleleft\triangleleft b, \triangleleft bb, bba, aab, ab\triangleright, b\triangleright\triangleright\}$
- Def. [Zalcstein 1972] strictly locally testable language $\text{Lang}(T) = \{w \mid \text{btiles}(w) \subseteq T\}$
- this paper:
  - use such languages to over-approximate $R^*(L)$
  - represent T by finite automaton A,
  - …constructed by completion
  - semantically label R by the partial algebra of A
  - …to transform the termination problem of R on L.
  - *sparse*: T is the set of tiles that occur in rhs of forward closures (overlap closures, resp.)
- application: Matchbox wins Termcomp 2019 for SRS

Sparse Tiling: Definition and Motivation

- Ex. the bordered 3-tiles of string $w = bbaab$ are $\text{btiles}_3(w) = \{\triangleleft\triangleleft b, \triangleleft bb, bba, aab, ab\triangleright, b\triangleright\triangleright\}$
- Def. [Zalcstein 1972] strictly locally testable language $\text{Lang}(T) = \{w \mid \text{btiles}(w) \subseteq T\}$
- this paper:
  - use such languages to over-approximate $R^*(L)$
  - represent T by finite automaton A,
  - ... constructed by completion
  - semantically label R by the partial algebra of A
  - ... to transform the termination problem of R on L.
  - *sparse*: T is the set of tiles that occur in rhs of forward closures (overlap closures, resp.)
- application: Matchbox wins Termcomp 2019 for SRS

Sparse Tiling: Definition and Motivation

- Ex. the bordered 3-tiles of string $w = bbaab$ are $\text{btiles}_3(w) = \{\lhd\lhd b, \lhd bb, bba, aab, ab\rhd, b\rhd\rhd\}$
- Def. [Zalcstein 1972] strictly locally testable language $\text{Lang}(T) = \{w \mid \text{btiles}(w) \subseteq T\}$
- this paper:
  - use such languages to over-approximate $R^*(L)$
  - represent T by finite automaton A,
  - ... constructed by completion
  - semantically label R by the partial algebra of A
  - ... to transform the termination problem of R on L.
  - *sparse*: T is the set of tiles that occur in rhs of forward closures (overlap closures, resp.)
- application: Matchbox wins Termcomp 2019 for SRS

Sparse Tiling: Definition and Motivation

- Ex. the bordered 3-tiles of string $w = bbaab$ are $\mathrm{btiles}_3(w) = \{\triangleleft\triangleleft b, \triangleleft bb, bba, aab, ab\triangleright, b\triangleright\triangleright\}$
- Def. [Zalcstein 1972] strictly locally testable language $\mathrm{Lang}(T) = \{w \mid \mathrm{btiles}(w) \subseteq T\}$
- this paper:
  - use such languages to over-approximate $R^*(L)$
  - represent T by finite automaton A,
  - ... constructed by completion
  - semantically label R by the partial algebra of A
  - ... to transform the termination problem of R on L.
  - *sparse*: T is the set of tiles that occur in rhs of forward closures (overlap closures, resp.)
- application: Matchbox wins Termcomp 2019 for SRS

Sparse Tiling: Definition and Motivation

- Ex. the bordered 3-tiles of string $w = bbaab$ are $\mathrm{btiles}_3(w) = \{\lhd\lhd b, \lhd bb, bba, aab, ab\rhd, b\rhd\rhd\}$
- Def. [Zalcstein 1972] strictly locally testable language $\mathrm{Lang}(T) = \{w \mid \mathrm{btiles}(w) \subseteq T\}$
- this paper:
  - use such languages to over-approximate $R^*(L)$
  - represent T by finite automaton A,
  - ...constructed by completion
  - semantically label R by the partial algebra of A
  - ...to transform the termination problem of R on L.
  - *sparse*: T is the set of tiles that occur in rhs of forward closures (overlap closures, resp.)
- application: Matchbox wins Termcomp 2019 for SRS

Right-hand Sides of Forward Closures

- Def. $\mathrm{RFC}(R) =$ smallest set $M \subseteq \Sigma^*$ with
  - (start) $\mathrm{rhs}(R) \subseteq M$
  - (inner step) $(l, r) \in R \wedge ulv \in M \Rightarrow urv \in M$
  - (right extension) $(l_1 l_2, r) \in R \wedge ul_1 \in M \Rightarrow ur \in M$
- Thm. (Dershowitz 1981)
  R terminates on $\Sigma^* \iff R$ terminates on $\mathrm{RFC}(R)$
- Ex. $\mathrm{RFC}(\{ab \to ba\}) = b^+ a$. Cor.: is terminating.
- Lemma: $\mathrm{RFC}(R) = (R \cup \mathrm{forw}(R))^*(\mathrm{rhs}(R))$ where
  $\mathrm{forw}(R) = \{l_1 \to_{\text{Suffix}} r \mid (l_1 l_2 \to r) \in R\}$.
- Ex. $\mathrm{RFC}(\{ab \to ba\}) = \{ab \to ba, a \to_{\text{Suffix}} ba\}^*(ba)$

Right-hand Sides of Forward Closures

- Def. $\mathrm{RFC}(R) =$ smallest set $M \subseteq \Sigma^*$ with
  - (start) $\mathrm{rhs}(R) \subseteq M$
  - (inner step) $(l, r) \in R \wedge ulv \in M \Rightarrow urv \in M$
  - (right extension) $(l_1 l_2, r) \in R \wedge ul_1 \in M \Rightarrow ur \in M$
- Thm. (Dershowitz 1981)
  R terminates on $\Sigma^* \iff R$ terminates on $\mathrm{RFC}(R)$
- Ex. $\mathrm{RFC}(\{ab \to ba\}) = b^+a$. Cor.: is terminating.
- Lemma: $\mathrm{RFC}(R) = (R \cup \mathrm{forw}(R))^*(\mathrm{rhs}(R))$ where
  $\mathrm{forw}(R) = \{l_1 \to_{\text{Suffix}} r \mid (l_1 l_2 \to r) \in R\}$.
- Ex. $\mathrm{RFC}(\{ab \to ba\}) = \{ab \to ba, a \to_{\text{Suffix}} ba\}^*(ba)$

Right-hand Sides of Forward Closures

- Def. $\mathrm{RFC}(R) =$ smallest set $M \subseteq \Sigma^*$ with
  - (start) $\mathrm{rhs}(R) \subseteq M$
  - (inner step) $(l, r) \in R \wedge ulv \in M \Rightarrow urv \in M$
  - (right extension) $(l_1 l_2, r) \in R \wedge ul_1 \in M \Rightarrow ur \in M$
- Thm. (Dershowitz 1981)
  R terminates on $\Sigma^* \iff R$ terminates on $\mathrm{RFC}(R)$
- Ex. $\mathrm{RFC}(\{ab \to ba\}) = b^+a$. Cor.: is terminating.
- Lemma: $\mathrm{RFC}(R) = (R \cup \mathrm{forw}(R))^*(\mathrm{rhs}(R))$ where
  $\mathrm{forw}(R) = \{l_1 \to_{\mathrm{Suffix}} r \mid (l_1 l_2 \to r) \in R\}$.
- Ex. $\mathrm{RFC}(\{ab \to ba\}) = \{ab \to ba, a \to_{\mathrm{Suffix}} ba\}^*(ba)$

Right-hand Sides of Forward Closures

- Def. $\mathrm{RFC}(R) =$ smallest set $M \subseteq \Sigma^*$ with
  - (start) $\mathrm{rhs}(R) \subseteq M$
  - (inner step) $(l, r) \in R \wedge ulv \in M \Rightarrow urv \in M$
  - (right extension) $(l_1 l_2, r) \in R \wedge ul_1 \in M \Rightarrow ur \in M$
- Thm. (Dershowitz 1981)
  R terminates on $\Sigma^* \iff R$ terminates on $\mathrm{RFC}(R)$
- Ex. $\mathrm{RFC}(\{ab \to ba\}) = b^+ a$. Cor.: is terminating.
- Lemma: $\mathrm{RFC}(R) = (R \cup \mathrm{forw}(R))^*(\mathrm{rhs}(R))$ where
  $\mathrm{forw}(R) = \{l_1 \to_{\mathrm{Suffix}} r \mid (l_1 l_2 \to r) \in R\}$.
- Ex. $\mathrm{RFC}(\{ab \to ba\}) = \{ab \to ba, a \to_{\mathrm{Suffix}} ba\}^*(ba)$

Right-hand Sides of Forward Closures

- Def. $\mathrm{RFC}(R) =$ smallest set $M \subseteq \Sigma^*$ with
  - (start) $\mathrm{rhs}(R) \subseteq M$
  - (inner step) $(l, r) \in R \wedge ulv \in M \Rightarrow urv \in M$
  - (right extension) $(l_1 l_2, r) \in R \wedge ul_1 \in M \Rightarrow ur \in M$
- Thm. (Dershowitz 1981)
  R terminates on $\Sigma^* \iff R$ terminates on $\mathrm{RFC}(R)$
- Ex. $\mathrm{RFC}(\{ab \to ba\}) = b^+a$.   Cor.: is terminating.
- Lemma: $\mathrm{RFC}(R) = (R \cup \mathrm{forw}(R))^*(\mathrm{rhs}(R))$ where
  $\mathrm{forw}(R) = \{l_1 \to_{\mathrm{Suffix}} r \mid (l_1 l_2 \to r) \in R\}$.
- Ex. $\mathrm{RFC}(\{ab \to ba\}) = \{ab \to ba, a \to_{\mathrm{Suffix}} ba\}^*(ba)$

Right-hand Sides of Forward Closures

- Def. $\mathrm{RFC}(R)$ = smallest set $M \subseteq \Sigma^*$ with
  - (start) $\mathrm{rhs}(R) \subseteq M$
  - (inner step) $(l, r) \in R \wedge ulv \in M \Rightarrow urv \in M$
  - (right extension) $(l_1 l_2, r) \in R \wedge ul_1 \in M \Rightarrow ur \in M$
- Thm. (Dershowitz 1981)
  R terminates on $\Sigma^* \iff R$ terminates on $\mathrm{RFC}(R)$
- Ex. $\mathrm{RFC}(\{ab \to ba\}) = b^+ a$. Cor.: is terminating.
- Lemma: $\mathrm{RFC}(R) = (R \cup \mathrm{forw}(R))^*(\mathrm{rhs}(R))$ where
  $\mathrm{forw}(R) = \{l_1 \to_{\mathrm{Suffix}} r \mid (l_1 l_2 \to r) \in R\}$.
- Ex. $\mathrm{RFC}(\{ab \to ba\}) = \{ab \to ba, a \to_{\mathrm{Suffix}} ba\}^*(ba)$

Right-hand Sides of Forward Closures

- Def. $\mathrm{RFC}(R)$ = smallest set $M \subseteq \Sigma^*$ with
  - (start) $\mathrm{rhs}(R) \subseteq M$
  - (inner step) $(l, r) \in R \wedge ulv \in M \Rightarrow urv \in M$
  - (right extension) $(l_1 l_2, r) \in R \wedge ul_1 \in M \Rightarrow ur \in M$
- Thm. (Dershowitz 1981)
  R terminates on $\Sigma^* \iff R$ terminates on $\mathrm{RFC}(R)$
- Ex. $\mathrm{RFC}(\{ab \to ba\}) = b^+ a$. Cor.: is terminating.
- Lemma: $\mathrm{RFC}(R) = (R \cup \mathrm{forw}(R))^*(\mathrm{rhs}(R))$ where
  $\mathrm{forw}(R) = \{l_1 \to_{\mathrm{Suffix}} r \mid (l_1 l_2 \to r) \in R\}$.
- Ex. $\mathrm{RFC}(\{ab \to ba\}) = \{ab \to ba, a \to_{\mathrm{Suffix}} ba\}^*(ba)$

Right-hand Sides of Forward Closures

- Def. $\text{RFC}(R) =$ smallest set $M \subseteq \Sigma^*$ with
  - (start) $\text{rhs}(R) \subseteq M$
  - (inner step) $(l, r) \in R \land ulv \in M \Rightarrow urv \in M$
  - (right extension) $(l_1 l_2, r) \in R \land ul_1 \in M \Rightarrow ur \in M$
- Thm. (Dershowitz 1981)
  R terminates on $\Sigma^* \iff R$ terminates on $\text{RFC}(R)$
- Ex. $\text{RFC}(\{ab \to ba\}) = b^+ a$. Cor.: is terminating.
- Lemma: $\text{RFC}(R) = (R \cup \text{forw}(R))^*(\text{rhs}(R))$ where
  $\text{forw}(R) = \{l_1 \to_{\text{Suffix}} r \mid (l_1 l_2 \to r) \in R\}$.
- Ex. $\text{RFC}(\{ab \to ba\}) = \{ab \to ba, a \to_{\text{Suffix}} ba\}^*(ba)$

Representing Sets of Tiles by Automata

- Def: the k-shift automaton
  (it remembers $k-1$ most recent letters read)
  alphabet $\Sigma \cup \{\triangleright\}$,
  states $\text{tiles}_{k-1}(\triangleleft^*\Sigma^*\triangleright^*)$, initial state $\triangleleft^{k-1}$, final state $\triangleright^{k-1}$,
  transitions: $p \xrightarrow{c}_A \text{Suffix}_{k-1}(pc)$
- represents set of k-tiles $\text{tiles}(A) := \{pc \mid p \xrightarrow{c}_A q\}$

- Ex. 2-shift automaton $A =$
  represents 2-tiles $\{\triangleleft a, \triangleleft b, ab, ac, bb, bc, c\triangleright\}$
  $\text{Lang}(A) = (a+b)b^*c$

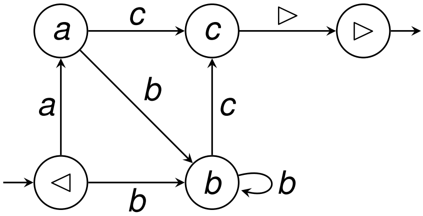
Representing Sets of Tiles by Automata

- Def: the k-shift automaton
  (it remembers $k-1$ most recent letters read)
  alphabet $\Sigma \cup \{\triangleright\}$,
  states $\text{tiles}_{k-1}(\triangleleft^*\Sigma^*\triangleright^*)$, initial state $\triangleleft^{k-1}$, final state $\triangleright^{k-1}$,
  transitions: $p \xrightarrow{c}_A \text{Suffix}_{k-1}(pc)$
- represents set of k-tiles $\text{tiles}(A) := \{pc \mid p \xrightarrow{c}_A q\}$

- Ex. 2-shift automaton $A =$
  represents 2-tiles $\{\triangleleft a, \triangleleft b, ab, ac, bb, bc, c\triangleright\}$
  $\text{Lang}(A) = (a+b)b^*c$

Representing Sets of Tiles by Automata

- Def: the k-shift automaton
  (it remembers $k-1$ most recent letters read)
  alphabet $\Sigma \cup \{\triangleright\}$,
  states $\text{tiles}_{k-1}(\triangleleft^*\Sigma^*\triangleright^*)$, initial state $\triangleleft^{k-1}$, final state $\triangleright^{k-1}$,
  transitions: $p \xrightarrow{c}_A \text{Suffix}_{k-1}(pc)$
- represents set of k-tiles $\text{tiles}(A) := \{pc \mid p \xrightarrow{c}_A q\}$
- Ex. 2-shift automaton $A =$



  represents 2-tiles $\{\triangleleft a, \triangleleft b, ab, ac, bb, bc, c\triangleright\}$
  $\text{Lang}(A) = (a+b)b^*c$

Rewrite Closure of Tiling Automata

- spec: given k-shift A, R over Σ, find k-shift A' over Σ s.t.
  - $\mathrm{Lang}(A) \subseteq \mathrm{Lang}(A')$
  - $u \in \mathrm{Lang}(A') \land u \to_R v \Rightarrow v \in \mathrm{Lang}(A')$
- implementation:
  when $(l, r) \in \mathrm{CC}_k(R)$ (right k-context closure)
  and $p \xrightarrow{l}_A q$,
  add transitions and states such that $p \xrightarrow{r}_A q$,
  until it stabilises
- by the k-shift property:
  - given p and r, the path $p \xrightarrow{r}_A q$ is fully determined,
    and it will indeed end in q
  - completion terminates since set of states is finite

Rewrite Closure of Tiling Automata

- spec: given k-shift A, R over Σ, find k-shift A' over Σ s.t.
  - $\text{Lang}(A) \subseteq \text{Lang}(A')$
  - $u \in \text{Lang}(A') \land u \to_R v \Rightarrow v \in \text{Lang}(A')$
- implementation:
  when $(l, r) \in \text{CC}_k(R)$ (right k-context closure)
  and $p \xrightarrow{l}_A q$,
  add transitions and states such that $p \xrightarrow{r}_A q$,
  until it stabilises
- by the k-shift property:
  - given p and r, the path $p \xrightarrow{r}_A q$ is fully determined and it will indeed end in q
  - completion terminates since set of states is finite

Rewrite Closure of Tiling Automata

- spec: given k-shift A, R over Σ, find k-shift A' over Σ s.t.
  - $\text{Lang}(A) \subseteq \text{Lang}(A')$
  - $u \in \text{Lang}(A') \wedge u \to_R v \Rightarrow v \in \text{Lang}(A')$
- implementation:
  when $(l, r) \in \text{CC}_k(R)$ (right k-context closure)
  and $p \xrightarrow{l}_A q$,
  add transitions and states such that $p \xrightarrow{r}_A q$,
  until it stabilises
- by the k-shift property:
  - given p and r, the path $p \xrightarrow{r}_A q$ is fully determined,
    and it will indeed end in q
  - completion terminates since set of states is finite

Rewrite Closure of Tiling Automata

- spec: given k-shift A, R over Σ, find k-shift A' over Σ s.t.
  - $\text{Lang}(A) \subseteq \text{Lang}(A')$
  - $u \in \text{Lang}(A') \wedge u \to_R v \Rightarrow v \in \text{Lang}(A')$
- implementation:
  when $(l, r) \in \text{CC}_k(R)$ (right k-context closure)
  and $p \xrightarrow{l}_A q$,
  add transitions and states such that $p \xrightarrow{r}_A q$,
  until it stabilises
- by the k-shift property:
  - given p and r, the path $p \xrightarrow{r}_A q$ is fully determined,
    and it will indeed end in q
  - completion terminates since set of states is finite

Rewrite Closure of Tiling Automata

- spec: given k-shift A, R over Σ, find k-shift A' over Σ s.t.
  - $\text{Lang}(A) \subseteq \text{Lang}(A')$
  - $u \in \text{Lang}(A') \wedge u \to_R v \Rightarrow v \in \text{Lang}(A')$
- implementation:
  when $(l, r) \in \text{CC}_k(R)$ (right k-context closure)
  and $p \xrightarrow{l}_A q$,
  add transitions and states such that $p \xrightarrow{r}_A q$,
  until it stabilises
- by the k-shift property:
  - given p and r, the path $p \xrightarrow{r}_A q$ is fully determined, and it will indeed end in q
  - completion terminates since set of states is finite

Rewrite Closure of Tiling Automata

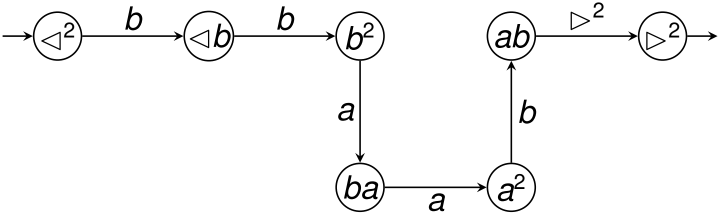
- spec: given k-shift A, R over Σ, find k-shift A' over Σ s.t.
  - $\text{Lang}(A) \subseteq \text{Lang}(A')$
  - $u \in \text{Lang}(A') \land u \to_R v \Rightarrow v \in \text{Lang}(A')$
- implementation:
  when $(l, r) \in \text{CC}_k(R)$ (right k-context closure)
  and $p \xrightarrow{l}_A q$,
  add transitions and states such that $p \xrightarrow{r}_A q$,
  until it stabilises
- by the k-shift property:
  - given p and r, the path $p \xrightarrow{r}_A q$ is fully determined, and it will indeed end in q
  - completion terminates since set of states is finite

Rewrite Closure of Tiling Automata

- spec: given k-shift A, R over Σ, find k-shift A' over Σ s.t.
  - $\text{Lang}(A) \subseteq \text{Lang}(A')$
  - $u \in \text{Lang}(A') \land u \to_R v \Rightarrow v \in \text{Lang}(A')$
- implementation:
  when $(l, r) \in \text{CC}_k(R)$ (right k-context closure)
  and $p \xrightarrow{l}_A q$,
  add transitions and states such that $p \xrightarrow{r}_A q$,
  until it stabilises
- by the k-shift property:
  - given p and r, the path $p \xrightarrow{r}_A q$ is fully determined, and it will indeed end in q
  - completion terminates since set of states is finite

Closure Example

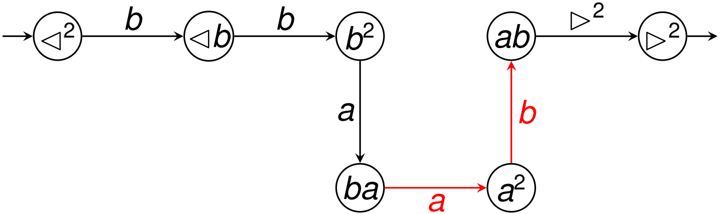
- for $R = \{ab^3 \to bbaab\}$,
  compute 3-shift approx. of $(R \cup \text{forw}(R))^*(\text{rhs}(R))$



- …this is the path for $\text{rhs}(R)$
- absent: $\triangleleft^2\triangleright, \triangleleft\triangleright^2, \triangleleft\Sigma\triangleright$, $\triangleleft a\Sigma, \triangleleft ba, \Sigma a\triangleright$, a^3

Closure Example

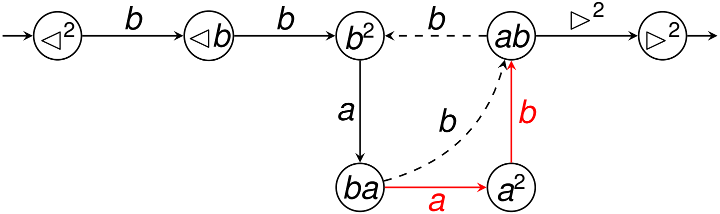
- for $R = \{ab^3 \to bbaab\}$,
  compute 3-shift approx. of $(R \cup \text{forw}(R))^*(\text{rhs}(R))$



- $\to$ a redex for $(ab \to_{\text{Suffix}} bbaab) \in \text{forw}(R)$
- absent: $\triangleleft^2\triangleright, \triangleleft\triangleright^2, \triangleleft\Sigma\triangleright,\quad \triangleleft a\Sigma, \triangleleft ba, \Sigma a\triangleright,\quad a^3$

Closure Example

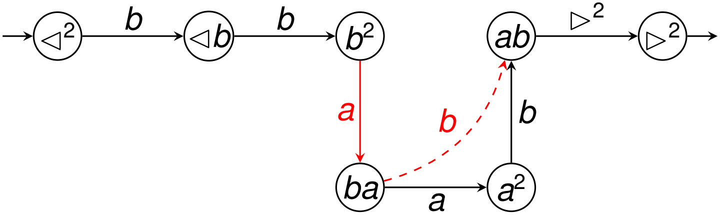
- for $R = \{ab^3 \to bbaab\}$,
  compute 3-shift approx. of $(R \cup \text{forw}(R))^*(\text{rhs}(R))$



- $\to$ a redex for $(ab \to_{\text{Suffix}} bbaab) \in \text{forw}(R)$
  dashed: new edges for corresponding reduct
- absent: $\lhd^2\rhd, \lhd\rhd^2, \lhd\Sigma\rhd, \quad \lhd a\Sigma, \lhd ba, \Sigma a\rhd, \quad a^3$

Closure Example

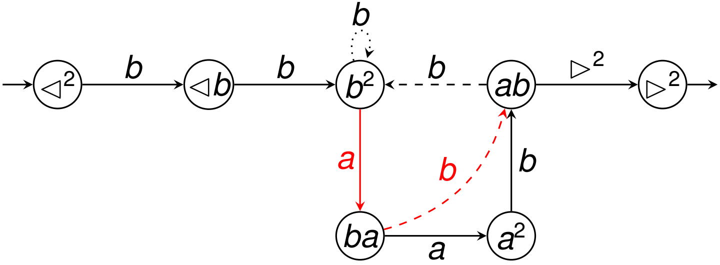
- for $R = \{ab^3 \to bbaab\}$,
  compute 3-shift approx. of $(R \cup \mathrm{forw}(R))^*(\mathrm{rhs}(R))$



- $\to$ a redex for $(ab \to_{\mathrm{Suffix}} bbaab) \in \mathrm{forw}(R)$
- absent: $\lhd^2\rhd, \lhd\rhd^2, \lhd\Sigma\rhd, \quad \lhd a\Sigma, \lhd ba, \Sigma a\rhd, \quad a^3$

Closure Example

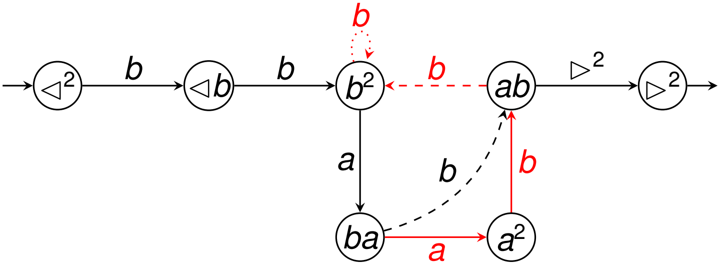
- for $R = \{ab^3 \to bbaab\}$,
  compute 3-shift approx. of $(R \cup \mathrm{forw}(R))^*(\mathrm{rhs}(R))$



- $\to$ a redex for $(ab \to_{\mathrm{Suffix}} bbaab) \in \mathrm{forw}(R)$
  dotted: new edge for corresponding reduct
- absent: $\lhd^2\rhd, \lhd\rhd^2, \lhd\Sigma\rhd, \quad \lhd a\Sigma, \lhd ba, \Sigma a\rhd, \quad a^3$

Closure Example

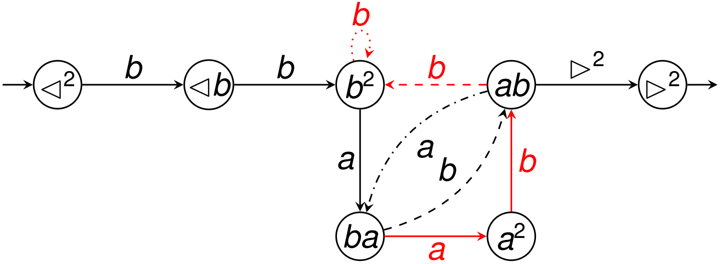
- for $R = \{ab^3 \to bbaab\}$,
  compute 3-shift approx. of $(R \cup \text{forw}(R))^*(\text{rhs}(R))$



- $\to$ a redex for $(ab^3a \to bbaaba) \in \text{CC}_1(R)$
- absent: $\triangleleft^2\triangleright, \triangleleft\triangleright^2, \triangleleft\Sigma\triangleright$, $\triangleleft a\Sigma, \triangleleft ba, \Sigma a\triangleright$, a^3

Closure Example

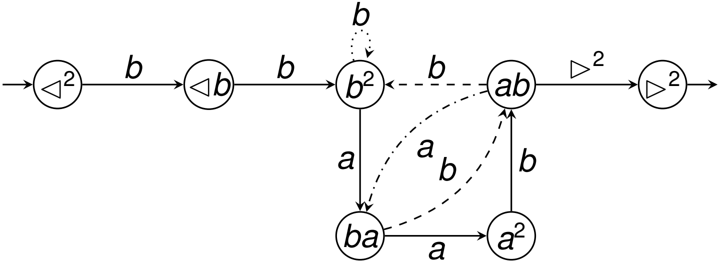
- for $R = \{ab^3 \to bbaab\}$,
  compute 3-shift approx. of $(R \cup \text{forw}(R))^*(\text{rhs}(R))$



- $\to$ a redex for $(ab^3a \to bbaaba) \in \text{CC}_1(R)$
  dash-dotted: new edge for corresponding reduct
- absent: $\lhd^2\rhd, \lhd\rhd^2, \lhd\Sigma\rhd, \quad \lhd a\Sigma, \lhd ba, \Sigma a\rhd, \quad a^3$

Closure Example

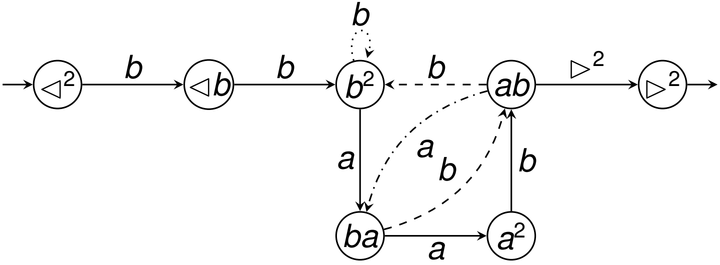
- for $R = \{ab^3 \to bbaab\}$,
  compute 3-shift approx. of $(R \cup \mathrm{forw}(R))^*(\mathrm{rhs}(R))$



- represents the set of tiles $T = \{\lhd\lhd b, \lhd bb, bba, bbb, baa, bab, aab, aba, abb, ab\rhd, b\rhd\rhd\}$.
- absent: $\lhd^2\rhd, \lhd\rhd^2, \lhd\Sigma\rhd$, $\lhd a\Sigma, \lhd ba, \Sigma a\rhd$, a^3

Closure Example

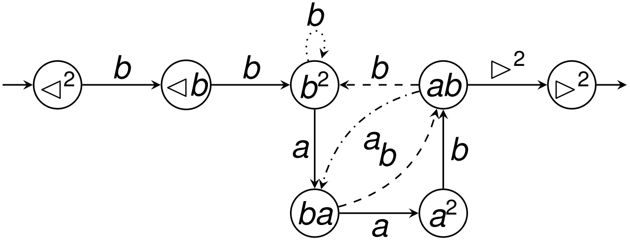
- for $R = \{ab^3 \to bbaab\}$,
  compute 3-shift approx. of $(R \cup \text{forw}(R))^*(\text{rhs}(R))$



- represents the set of tiles $T = \{\lhd\lhd b, \lhd bb, bba, bbb, baa, bab, aab, aba, abb, ab\rhd, b\rhd\rhd\}$.
- absent: $\lhd^2\rhd, \lhd\rhd^2, \lhd\Sigma\rhd, \quad \lhd a\Sigma, \lhd ba, \Sigma a\rhd, \quad a^3$

Semantic Labelling

- for $R = \{ab^3 \to bbaab\}$,



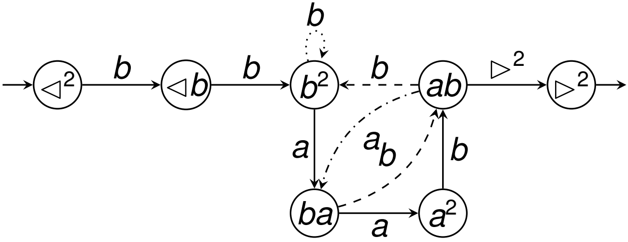
- semantically labelled R is $R' =$

$$bba, bab, abb, b^3, bbx, bxy \to b^3, b^3, bba, baa, aab, abx, bxy$$
$$baa, aab, abb, b^3, bbx, bxy \to bab, abb, bba, baa, aab, abx, bxy$$
$$aba, bab, abb, b^3, bbx, bxy \to abb, b^3, bba, baa, aab, abx, bxy$$

- SN(R') by weights $b^3 \mapsto 8$, $bab \mapsto 4$, $abb \mapsto 3$, $bba \mapsto 3$

Semantic Labelling

- for $R = \{ab^3 \to bbaab\}$,



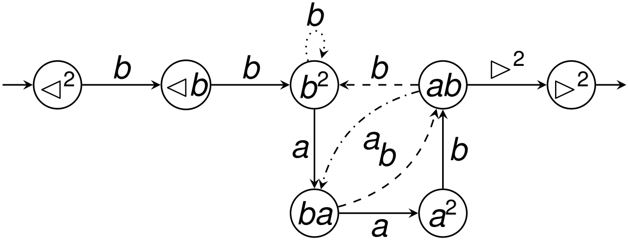
- semantically labelled R is $R' =$

$$bba, bab, abb, b^3, bbx, bxy \to b^3, b^3, bba, baa, aab, abx, bxy$$
$$baa, aab, abb, b^3, bbx, bxy \to bab, abb, bba, baa, aab, abx, bxy$$
$$aba, bab, abb, b^3, bbx, bxy \to abb, b^3, bba, baa, aab, abx, bxy$$

- SN(R') by weights $b^3 \mapsto 8$, $bab \mapsto 4$, $abb \mapsto 3$, $bba \mapsto 3$

Semantic Labelling

- for $R = \{ab^3 \to bbaab\}$,



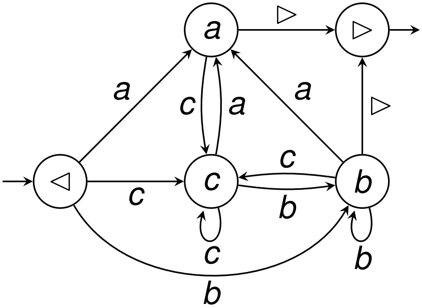
- semantically labelled R is $R' =$

$$bba, bab, abb, b^3, bbx, bxy \to b^3, b^3, bba, baa, aab, abx, bxy$$
$$baa, aab, abb, b^3, bbx, bxy \to bab, abb, bba, baa, aab, abx, bxy$$
$$aba, bab, abb, b^3, bbx, bxy \to abb, b^3, bba, baa, aab, abx, bxy$$

- SN(R') by weights $b^3 \mapsto 8, bab \mapsto 4, abb \mapsto 3, bba \mapsto 3$

Removing unreachable rules (Prop. 5.3)

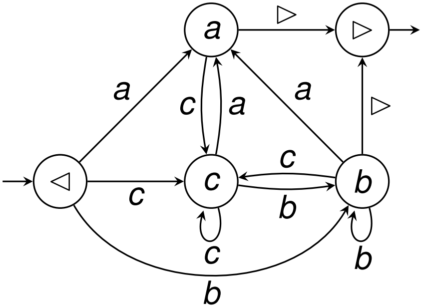
- Ex. 5.5 $R = \{ab \to bca, bc \to cbb, ba \to acb\}$.



- $\text{btiled}_T(ab \to bca) = \emptyset$ implies
  $\text{SN}(R) \iff \text{SN}(bc \to cbb, ba \to acb)$.
- we remove rule $ab \to bca$, even though A still contains redexes for $a \to_{\text{Suffix}} bca$.

Removing unreachable rules (Prop. 5.3)

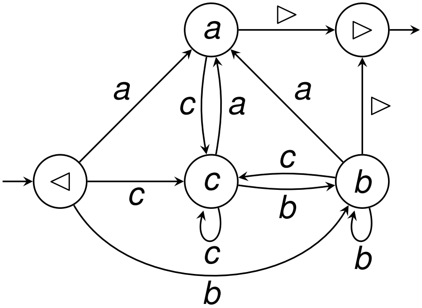
- Ex. 5.5 $R = \{ab \to bca, bc \to cbb, ba \to acb\}$.



- $\text{btiled}_T(ab \to bca) = \emptyset$ implies
  $\text{SN}(R) \iff \text{SN}(bc \to cbb, ba \to acb)$.
- we remove rule $ab \to bca$, even though A still contains redexes for $a \to_{\text{Suffix}} bca$.

Removing unreachable rules (Prop. 5.3)

- Ex. 5.5 $R = \{ab \to bca, bc \to cbb, ba \to acb\}$.



- $\text{btiled}_T(ab \to bca) = \emptyset$ implies
  $\text{SN}(R) \iff \text{SN}(bc \to cbb, ba \to acb)$.
- we remove rule $ab \to bca$, even though A still contains redexes for $a \to_{\text{Suffix}} bca$.

Killer example: $a^2 b^2 \to b^3 a^3$

- Theorem: each paper on SRS termination contains a termination proof for Zantema's ($\approx$ 1993) problem
- Fact: as *z001*, it appears in the Termination Problems Data Base since the beginning of time ($=$ 2003)
- tiling for RFC; with semantic labelling (All), rule removal (Rem), weights (W); showing $(|R|, |\Sigma|)$ for each step:

$$(1,2) \xrightarrow[\text{All}]{\text{RFC}_2} (4,4) \xrightarrow[\text{Rem}]{\text{RFC}_5} (3,4) \xrightarrow[\text{All}]{\text{RFC}_2} (12,8) \xrightarrow[\text{All}]{\text{RFC}_3} (105,26) \xrightarrow{\text{W}} (60,26)$$

$$\xrightarrow[\text{Rem}]{\text{RFC}_5} (37,26) \xrightarrow[\text{All}]{\text{RFC}_2} (97,44) \xrightarrow{\text{W}} (65,43) \xrightarrow[\text{Rem}]{\text{RFC}_5} (36,43) \xrightarrow{\text{W}} (28,43)$$

$$\xrightarrow[\text{All}]{\text{RFC}_2} (86,68) \xrightarrow{\text{W}} (50,62) \xrightarrow[\text{All}]{\text{RFC}_3} (246,128) \xrightarrow{\text{W}} (42,84)$$

$$\xrightarrow[\text{Rem}]{\text{RFC}_5} (2,44) \xrightarrow{\text{W}} (0,0)$$

Killer example: $a^2 b^2 \to b^3 a^3$

- Theorem: each paper on SRS termination contains a termination proof for Zantema's ($\approx$ 1993) problem
- Fact: as *z001*, it appears in the Termination Problems Data Base since the beginning of time ($=$ 2003)
- tiling for RFC; with semantic labelling (All), rule removal (Rem), weights (W); showing $(|R|, |\Sigma|)$ for each step:

$$(1,2) \xrightarrow[\text{All}]{\text{RFC}_2} (4,4) \xrightarrow[\text{Rem}]{\text{RFC}_5} (3,4) \xrightarrow[\text{All}]{\text{RFC}_2} (12,8) \xrightarrow[\text{All}]{\text{RFC}_3} (105,26) \xrightarrow{\text{W}} (60,26)$$

$$\xrightarrow[\text{Rem}]{\text{RFC}_5} (37,26) \xrightarrow[\text{All}]{\text{RFC}_2} (97,44) \xrightarrow{\text{W}} (65,43) \xrightarrow[\text{Rem}]{\text{RFC}_5} (36,43) \xrightarrow{\text{W}} (28,43)$$

$$\xrightarrow[\text{All}]{\text{RFC}_2} (86,68) \xrightarrow{\text{W}} (50,62) \xrightarrow[\text{All}]{\text{RFC}_3} (246,128) \xrightarrow{\text{W}} (42,84)$$

$$\xrightarrow[\text{Rem}]{\text{RFC}_5} (2,44) \xrightarrow{\text{W}} (0,0)$$

Killer example: $a^2b^2 \to b^3a^3$

- Theorem: each paper on SRS termination contains a termination proof for Zantema's ($\approx$ 1993) problem
- Fact: as *z001*, it appears in the Termination Problems Data Base since the beginning of time ($=$ 2003)
- tiling for RFC; with semantic labelling (All), rule removal (Rem), weights (W); showing $(|R|, |\Sigma|)$ for each step:

$$(1,2) \xrightarrow[\text{All}]{\text{RFC}_2} (4,4) \xrightarrow[\text{Rem}]{\text{RFC}_5} (3,4) \xrightarrow[\text{All}]{\text{RFC}_2} (12,8) \xrightarrow[\text{All}]{\text{RFC}_3} (105,26) \xrightarrow{\text{W}} (60,26)$$

$$\xrightarrow[\text{Rem}]{\text{RFC}_5} (37,26) \xrightarrow[\text{All}]{\text{RFC}_2} (97,44) \xrightarrow{\text{W}} (65,43) \xrightarrow[\text{Rem}]{\text{RFC}_5} (36,43) \xrightarrow{\text{W}} (28,43)$$

$$\xrightarrow[\text{All}]{\text{RFC}_2} (86,68) \xrightarrow{\text{W}} (50,62) \xrightarrow[\text{All}]{\text{RFC}_3} (246,128) \xrightarrow{\text{W}} (42,84)$$

$$\xrightarrow[\text{Rem}]{\text{RFC}_5} (2,44) \xrightarrow{\text{W}} (0,0)$$

Overlap Closures and Relative Termination

- Def: R terminates relative to S, notation: $\mathrm{SN}(R/S)$, if there is no $(R \cup S)$-derivation with infinitely many R steps.
  Ex: $\mathrm{SN}(aa \to aba / a \to aba)$.
- (recap) $\mathrm{SN}(R)$ iff $\mathrm{SN}(R)$ on $\mathrm{RFC}(R)$.
- (Ex. 6.1) $\mathrm{SN}(R/S)$ on $\mathrm{RFC}(R \cup S) \not\Rightarrow \mathrm{SN}(R/S)$.
  $R = \{ab \to a\}, S = \{c \to bc\}, \mathrm{RFC}(R \cup S) = a \cup b^+c$.
  But $abc \to_R ac \to_S abc$.
- Thm 6.7 $\mathrm{SN}(R/S)$ iff $\mathrm{SN}(R/S)$ on $\mathrm{ROC}(R \cup S)$.
  using right-hand sides of *overlap* closures
- apply left-recursive characterisation of ROC (overlap closure with rule) (see Appendix of paper).
- interesting case: (Cor 7.1.5)
  if $tx \in S$ and $yv \in S$ and $(xwy, z) \in R$, then $tzv \in S$

Overlap Closures and Relative Termination

- Def: R terminates relative to S, notation: SN(R/S), if there is no ($R \cup S$)-derivation with infinitely many R steps.
  Ex: SN($aa \to aba / a \to aba$).
- (recap) SN(R) iff SN(R) on RFC(R).
- (Ex. 6.1) SN(R/S) on RFC($R \cup S$) $\not\Rightarrow$ SN(R/S).
  $R = \{ab \to a\}, S = \{c \to bc\}$, RFC($R \cup S$) $= a \cup b^+c$.
  But $abc \to_R ac \to_S abc$.
- Thm 6.7 SN(R/S) iff SN(R/S) on ROC($R \cup S$).
  using right-hand sides of *overlap* closures
- apply left-recursive characterisation of ROC (overlap closure with rule) (see Appendix of paper).
- interesting case: (Cor 7.1.5)
  if $tx \in S$ and $yv \in S$ and $(xwy, z) \in R$, then $tzv \in S$

Overlap Closures and Relative Termination

- Def: R terminates relative to S, notation: SN(R/S), if there is no $(R \cup S)$-derivation with infinitely many R steps.
  Ex: SN($aa \to aba / a \to aba$).
- (recap) SN(R) iff SN(R) on RFC(R).
- (Ex. 6.1) SN(R/S) on RFC($R \cup S$) $\not\Rightarrow$ SN(R/S).
  $R = \{ab \to a\}, S = \{c \to bc\}, \mathrm{RFC}(R \cup S) = a \cup b^+ c$.
  But $abc \to_R ac \to_S abc$.
- Thm 6.7 SN(R/S) iff SN(R/S) on ROC($R \cup S$).
  using right-hand sides of *overlap* closures
- apply left-recursive characterisation of ROC (overlap closure with rule) (see Appendix of paper).
- interesting case: (Cor 7.1.5)
  if $tx \in S$ and $yv \in S$ and $(xwy, z) \in R$, then $tzv \in S$

Overlap Closures and Relative Termination

- Def: R terminates relative to S, notation: $\mathrm{SN}(R/S)$, if there is no $(R \cup S)$-derivation with infinitely many R steps.
  Ex: $\mathrm{SN}(aa \to aba / a \to aba)$.
- (recap) $\mathrm{SN}(R)$ iff $\mathrm{SN}(R)$ on $\mathrm{RFC}(R)$.
- (Ex. 6.1) $\mathrm{SN}(R/S)$ on $\mathrm{RFC}(R \cup S) \not\Rightarrow \mathrm{SN}(R/S)$.
  $R = \{ab \to a\}, S = \{c \to bc\}, \mathrm{RFC}(R \cup S) = a \cup b^+c$.
  But $abc \to_R ac \to_S abc$.
- Thm 6.7 $\mathrm{SN}(R/S)$ iff $\mathrm{SN}(R/S)$ on $\mathrm{ROC}(R \cup S)$.
  using right-hand sides of *overlap* closures
- apply left-recursive characterisation of ROC (overlap closure with rule) (see Appendix of paper).
- interesting case: (Cor 7.1.5)
  if $tx \in S$ and $yv \in S$ and $(xwy, z) \in R$, then $tzv \in S$

Overlap Closures and Relative Termination

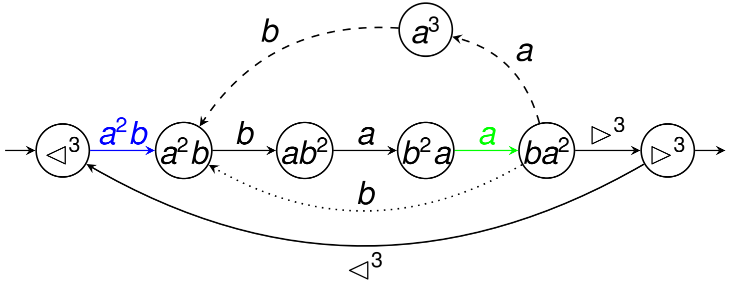
- Def: R terminates relative to S, notation: $\mathrm{SN}(R/S)$, if there is no $(R \cup S)$-derivation with infinitely many R steps.
  Ex: $\mathrm{SN}(aa \to aba/a \to aba)$.
- (recap) $\mathrm{SN}(R)$ iff $\mathrm{SN}(R)$ on $\mathrm{RFC}(R)$.
- (Ex. 6.1) $\mathrm{SN}(R/S)$ on $\mathrm{RFC}(R \cup S) \not\Rightarrow \mathrm{SN}(R/S)$.
  $R = \{ab \to a\}, S = \{c \to bc\}, \mathrm{RFC}(R \cup S) = a \cup b^+c$.
  But $abc \to_R ac \to_S abc$.
- Thm 6.7 $\mathrm{SN}(R/S)$ iff $\mathrm{SN}(R/S)$ on $\mathrm{ROC}(R \cup S)$.
  using right-hand sides of *overlap* closures
- apply left-recursive characterisation of ROC (overlap closure with rule) (see Appendix of paper).
- interesting case: (Cor 7.1.5)
  if $tx \in S$ and $yv \in S$ and $(xwy, z) \in R$, then $tzv \in S$

Overlap Closures and Relative Termination

- Def: R terminates relative to S, notation: SN(R/S), if there is no $(R \cup S)$-derivation with infinitely many R steps.
  Ex: SN($aa \to aba / a \to aba$).
- (recap) SN(R) iff SN(R) on RFC(R).
- (Ex. 6.1) SN(R/S) on RFC($R \cup S$) $\not\Rightarrow$ SN(R/S).
  $R = \{ab \to a\}, S = \{c \to bc\}, \mathrm{RFC}(R \cup S) = a \cup b^+c$.
  But $abc \to_R ac \to_S abc$.
- Thm 6.7 SN(R/S) iff SN(R/S) on ROC($R \cup S$).
  using right-hand sides of *overlap* closures
- apply left-recursive characterisation of ROC (overlap closure with rule) (see Appendix of paper).
- interesting case: (Cor 7.1.5)
  if $tx \in S$ and $yv \in S$ and $(xwy, z) \in R$, then $tzv \in S$

Example: Tiling for Overlap Closures

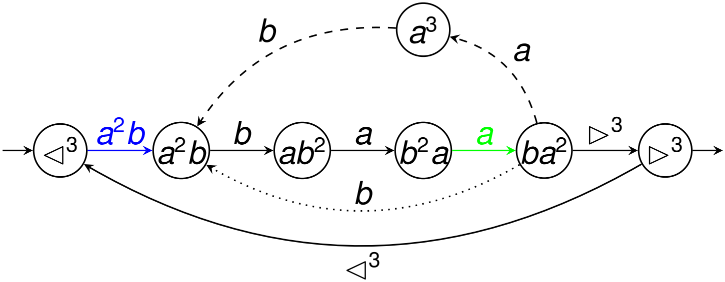
- 4-tiles for ROC(R), for $R = \{a^3 \to a^2 b^2 a^2\}$.



- if $tx \in S$ and $yv \in S$ and $(xwy, z) \in R$, then $tzv \in S$
  x is path to final state (since $x \in$ Suffix(S))
  y is path from initial state (since $y \in$ Prefix(S))
  use rewrite rule with border letters: $x \rhd^{k-1} \lhd^{k-1} y \to z$
  Ex: $aaa \cdot ab \to a^2b^2a^2 \cdot ab$, reduct needs dashed edges

Example: Tiling for Overlap Closures

- 4-tiles for ROC(R), for $R = \{a^3 \to a^2b^2a^2\}$.



- if $tx \in S$ and $yv \in S$ and $(xwy, z) \in R$, then $tzv \in S$
  x is path to final state (since $x \in \text{Suffix}(S)$)
  y is path from initial state (since $y \in \text{Prefix}(S)$)
  use rewrite rule with border letters: $x \rhd^{k-1} \lhd^{k-1} y \to z$
  Ex: $aaa \cdot ab \to a^2b^2a^2 \cdot ab$, reduct needs dashed edges

Implementation, Experiments, Questions

- implemented as part of termination prover
  `https://gitlab.imn.htwk-leipzig.de/waldmann/pure-matchbox`
- performance, including *Termcomp 2019* (SRS)

Relative		matrices no	matrices yes
tiling	no	1	72
tiling	yes	176	*225*

Standard		MB, DP, matr. none	MB, DP, matr. all
tiling	no	100	1122
tiling	yes	512	*1133*

- ? better proof search strategy for SRS Standard
- ? sparse tiling for TRS (RFC needs linearity)
- ? relation between matchbounds and tiling
- ? relation between tilings of different widths

Implementation, Experiments, Questions

- implemented as part of termination prover
  `https://gitlab.imn.htwk-leipzig.de/waldmann/pure-matchbox`
- performance, including *Termcomp 2019* (SRS)

Relative		matrices	
		no	yes
tiling	no	1	72
	yes	176	*225*

Standard		MB, DP, matr.	
		none	all
tiling	no	100	1122
	yes	512	*1133*

- ? better proof search strategy for SRS Standard
- ? sparse tiling for TRS (RFC needs linearity)
- ? relation between matchbounds and tiling
- ? relation between tilings of different widths

Implementation, Experiments, Questions

- implemented as part of termination prover `https://gitlab.imn.htwk-leipzig.de/waldmann/pure-matchbox`
- performance, including *Termcomp 2019* (SRS)

Relative		matrices no	matrices yes
tiling	no	1	72
tiling	yes	176	*225*

Standard		MB, DP, matr. none	MB, DP, matr. all
tiling	no	100	1122
tiling	yes	512	*1133*

- ? better proof search strategy for SRS Standard
- ? sparse tiling for TRS (RFC needs linearity)
- ? relation between matchbounds and tiling
- ? relation between tilings of different widths

Implementation, Experiments, Questions

- implemented as part of termination prover
  `https://gitlab.imn.htwk-leipzig.de/waldmann/pure-matchbox`
- performance, including *Termcomp 2019* (SRS)

Relative		matrices no	matrices yes
tiling	no	1	72
tiling	yes	176	*225*

Standard		MB, DP, matr. none	MB, DP, matr. all
tiling	no	100	1122
tiling	yes	512	*1133*

- ? better proof search strategy for SRS Standard
- ? sparse tiling for TRS (RFC needs linearity)
- ? relation between matchbounds and tiling
- ? relation between tilings of different widths

Implementation, Experiments, Questions

- implemented as part of termination prover `https://gitlab.imn.htwk-leipzig.de/waldmann/pure-matchbox`
- performance, including *Termcomp 2019* (SRS)

Relative		matrices no	matrices yes
tiling	no	1	72
tiling	yes	176	*225*

Standard		MB, DP, matr. none	MB, DP, matr. all
tiling	no	100	1122
tiling	yes	512	*1133*

- ? better proof search strategy for SRS Standard
- ? sparse tiling for TRS (RFC needs linearity)
- ? relation between matchbounds and tiling
- ? relation between tilings of different widths

Implementation, Experiments, Questions

- implemented as part of termination prover `https://gitlab.imn.htwk-leipzig.de/waldmann/pure-matchbox`
- performance, including *Termcomp 2019* (SRS)

Relative		matrices no	matrices yes
tiling	no	1	72
tiling	yes	176	*225*

Standard		MB, DP, matr. none	MB, DP, matr. all
tiling	no	100	1122
tiling	yes	512	*1133*

- ? better proof search strategy for SRS Standard
- ? sparse tiling for TRS (RFC needs linearity)
- ? relation between matchbounds and tiling
- ? relation between tilings of different widths