Proving Non-Joinability using Weakly Monotone Algebras

Bertram Felgenhauer (AoE)
Johannes Waldmann (HTWK Leipzig)

IWC 2019

Motivation

- Def: *peak*: $s\ {}^*\!\leftarrow \cdot \rightarrow^* t$, *joinable*: $s \rightarrow^* \cdot {}^*\!\leftarrow t$
  confluent: each peak is joinable
- non-joinable: $\rightarrow^*(s) \cap \rightarrow^*(t) = \varnothing$.
  If $\rightarrow$ is non-terminating, then $\rightarrow^*(s), \rightarrow^*(t)$ can be infinite.
- . . . and need to be described in some finite way,
  e.g., as finite automata $A \supseteq \rightarrow^*(s)$, $B \supseteq \rightarrow^*(t)$.
  then check emptiness of $A \cap B$ (Zankl et al., 2011)
- this paper:
  - use weighted automata A, B,
    representing weakly monotone algebras,
  - such that Kronecker product algebra (represents
    $x \mapsto A(x) \cdot B(x)$) has bounded weights
  - such that bound is less than $A(s) \cdot B(t)$.

Motivation

- Def: *peak*: $s \;{}^*\!\!\leftarrow \cdot \to^* t$, *joinable*: $s \to^* \cdot \;{}^*\!\!\leftarrow t$
  confluent: each peak is joinable
- non-joinable: $\to^*(s) \cap \to^*(t) = \emptyset$.
  If $\to$ is non-terminating, then $\to^*(s), \to^*(t)$ can be infinite.
- …and need to be described in some finite way,
  e.g., as finite automata $A \supseteq \to^*(s)$, $B \supseteq \to^*(t)$.
  then check emptiness of $A \cap B$ (Zankl et al., 2011)
- this paper:
  - use weighted automata A, B,
    representing weakly monotone algebras,
  - such that Kronecker product algebra (represents
    $x \mapsto A(x) \cdot B(x)$) has bounded weights
  - such that bound is less than $A(s) \cdot B(t)$.

Motivation

- Def: *peak*: $s \;{}^*\!\!\leftarrow \cdot \rightarrow^* t$, *joinable*: $s \rightarrow^* \cdot {}^*\!\!\leftarrow t$
  confluent: each peak is joinable
- non-joinable: $\rightarrow^*(s) \cap \rightarrow^*(t) = \emptyset$.
  If $\rightarrow$ is non-terminating, then $\rightarrow^*(s), \rightarrow^*(t)$ can be infinite.
- …and need to be described in some finite way,
  e.g., as finite automata $A \supseteq \rightarrow^*(s)$, $B \supseteq \rightarrow^*(t)$.
  then check emptiness of $A \cap B$ (Zankl et al., 2011)
- this paper:
  - use weighted automata A, B,
    representing weakly monotone algebras,
  - such that Kronecker product algebra (represents
    $x \mapsto A(x) \cdot B(x)$) has bounded weights
  - such that bound is less than $A(s) \cdot B(t)$.

Motivation

- Def: *peak*: $s \; {}^*\!\!\leftarrow \cdot \rightarrow^* \; t$, *joinable*: $s \rightarrow^* \cdot \; {}^*\!\!\leftarrow t$
  confluent: each peak is joinable
- non-joinable: $\rightarrow^*(s) \cap \rightarrow^*(t) = \varnothing$.
  If $\rightarrow$ is non-terminating, then $\rightarrow^*(s), \rightarrow^*(t)$ can be infinite.
- ...and need to be described in some finite way,
  e.g., as finite automata $A \supseteq \rightarrow^*(s)$, $B \supseteq \rightarrow^*(t)$.
  then check emptiness of $A \cap B$ (Zankl et al., 2011)
- this paper:
  - use weighted automata A, B,
    representing weakly monotone algebras,
  - such that Kronecker product algebra (represents
    $x \mapsto A(x) \cdot B(x)$) has bounded weights
  - such that bound is less than $A(s) \cdot B(t)$.

Motivation

- Def: *peak*: $s \;{}^*\!\!\leftarrow \cdot \rightarrow^* t$, *joinable*: $s \rightarrow^* \cdot {}^*\!\!\leftarrow t$
  confluent: each peak is joinable
- non-joinable: $\rightarrow^*(s) \cap \rightarrow^*(t) = \emptyset$.
  If $\rightarrow$ is non-terminating, then $\rightarrow^*(s), \rightarrow^*(t)$ can be infinite.
- ... and need to be described in some finite way,
  e.g., as finite automata $A \supseteq \rightarrow^*(s)$, $B \supseteq \rightarrow^*(t)$.
  then check emptiness of $A \cap B$ (Zankl et al., 2011)
- this paper:
  - use weighted automata A, B,
    representing weakly monotone algebras,
  - such that Kronecker product algebra (represents $x \mapsto A(x) \cdot B(x)$) has bounded weights
  - such that bound is less than $A(s) \cdot B(t)$.

Motivation

- Def: *peak*: $s \; {}^*\!\!\leftarrow \cdot \rightarrow^* t$, *joinable*: $s \rightarrow^* \cdot \; {}^*\!\!\leftarrow t$
  confluent: each peak is joinable
- non-joinable: $\rightarrow^*(s) \cap \rightarrow^*(t) = \varnothing$.
  If $\rightarrow$ is non-terminating, then $\rightarrow^*(s), \rightarrow^*(t)$ can be infinite.
- ...and need to be described in some finite way,
  e.g., as finite automata $A \supseteq \rightarrow^*(s)$, $B \supseteq \rightarrow^*(t)$.
  then check emptiness of $A \cap B$ (Zankl et al., 2011)
- this paper:
  - use weighted automata A, B,
    representing weakly monotone algebras,
  - such that Kronecker product algebra (represents $x \mapsto A(x) \cdot B(x)$) has bounded weights
  - such that bound is less than $A(s) \cdot B(t)$.

Motivation

- Def: *peak*: $s \,{}^*\!\!\leftarrow \cdot \rightarrow^* t$, *joinable*: $s \rightarrow^* \cdot \,{}^*\!\!\leftarrow t$
  confluent: each peak is joinable
- non-joinable: $\rightarrow^*(s) \cap \rightarrow^*(t) = \emptyset$.
  If $\rightarrow$ is non-terminating, then $\rightarrow^*(s), \rightarrow^*(t)$ can be infinite.
- …and need to be described in some finite way,
  e.g., as finite automata $A \supseteq \rightarrow^*(s)$, $B \supseteq \rightarrow^*(t)$.
  then check emptiness of $A \cap B$ (Zankl et al., 2011)
- this paper:
  - use weighted automata A, B,
    representing weakly monotone algebras,
  - such that Kronecker product algebra (represents $x \mapsto A(x) \cdot B(x)$) has bounded weights
  - such that bound is less than $A(s) \cdot B(t)$.

Example (Ex. 1)

- non-joinability of ag, bh with respect to $\mathcal{R} = \{g \to ag, g \to i, h \to bh, h \to i, i \to abi, ab \to ba, ba \to ab\}$
- algebras $A : s \mapsto \#_a(s) - \#_b(s)$, $B : s \mapsto \#_b(s) - \#_a(s)$,
  - for $s \in \to_{\mathcal{R}}^*(ag) : 1 \leq A(s)$ — note: $(h \to bh)$ not usable
  - for $s \in \to_{\mathcal{R}}^*(bh) : 1 \leq B(s)$ — note: $(g \to ag)$ not usable
  - for all $s : A(ag) + B(bh) = 2 \not\leq 0 = A(s) + B(s)$
- cannot separate $\to_{\mathcal{R}}^*(ag)$ from $\to_{\mathcal{R}}^*(bh)$ with regular languages since:

$$\to_{\mathcal{R}}^*(ag) \supseteq \{a^n b^m i \mid n > m\}, \quad \to_{\mathcal{R}}^*(bh) \supseteq \{a^n b^m i \mid n < m\}$$

- represent A, B as arctically $(\{-\infty\} \cup \mathbb{Z}, \max, +)$ weighted automata, with one state each.
  Encode non-usability by $A(h) = -\infty$, $B(g) = -\infty$.

Example (Ex. 1)

- non-joinability of ag, bh with respect to $\mathcal{R} = \{g \to ag, g \to i, h \to bh, h \to i, i \to abi, ab \to ba, ba \to ab\}$
- algebras $A : s \mapsto \#_a(s) - \#_b(s)$, $B : s \mapsto \#_b(s) - \#_a(s)$,
  - for $s \in \to^*_{\mathcal{R}}(ag) : 1 \leq A(s)$ note: $(h \to bh)$ not usable
  - for $s \in \to^*_{\mathcal{R}}(bh) : 1 \leq B(s)$ note: $(g \to ag)$ not usable
  - for all $s : A(ag) + B(bh) = 2 \not\leq 0 = A(s) + B(s)$
- cannot separate $\to^*_{\mathcal{R}}(ag)$ from $\to^*_{\mathcal{R}}(bh)$ with regular languages since:

$$\to^*_{\mathcal{R}}(ag) \supseteq \{a^n b^m i \mid n > m\}, \quad \to^*_{\mathcal{R}}(bh) \supseteq \{a^n b^m i \mid n < m\}$$

- represent A, B as arctically $(\{-\infty\} \cup \mathbb{Z}, \max, +)$ weighted automata, with one state each.
  Encode non-usability by $A(h) = -\infty, B(g) = -\infty$.

Example (Ex. 1)

- non-joinability of ag, bh with respect to $\mathcal{R} = \{g \to ag, g \to i, h \to bh, h \to i, i \to abi, ab \to ba, ba \to ab\}$
- algebras $A : s \mapsto \#_a(s) - \#_b(s)$, $B : s \mapsto \#_b(s) - \#_a(s)$,
  - for $s \in \to^*_{\mathcal{R}}(ag) : 1 \leq A(s)$ note: $(h \to bh)$ not usable
  - for $s \in \to^*_{\mathcal{R}}(bh) : 1 \leq B(s)$ note: $(g \to ag)$ not usable
  - for all $s : A(ag) + B(bh) = 2 \not\leq 0 = A(s) + B(s)$
- cannot separate $\to^*_{\mathcal{R}}(ag)$ from $\to^*_{\mathcal{R}}(bh)$ with regular languages since:

$$\to^*_{\mathcal{R}}(ag) \supseteq \{a^n b^m i \mid n > m\}, \quad \to^*_{\mathcal{R}}(bh) \supseteq \{a^n b^m i \mid n < m\}$$

- represent A, B as arctically $(\{-\infty\} \cup \mathbb{Z}, \max, +)$ weighted automata, with one state each.
  Encode non-usability by $A(h) = -\infty$, $B(g) = -\infty$.

Example (Ex. 1)

- non-joinability of ag, bh with respect to $\mathcal{R} = \{g \to ag, g \to i, h \to bh, h \to i, i \to abi, ab \to ba, ba \to ab\}$
- algebras $A : s \mapsto \#_a(s) - \#_b(s)$, $B : s \mapsto \#_b(s) - \#_a(s)$,
  - for $s \in \to^*_{\mathcal{R}}(ag) : 1 \leq A(s)$ note: $(h \to bh)$ not usable
  - for $s \in \to^*_{\mathcal{R}}(bh) : 1 \leq B(s)$ note: $(g \to ag)$ not usable
  - for all $s : A(ag) + B(bh) = 2 \not\leq 0 = A(s) + B(s)$
- cannot separate $\to^*_{\mathcal{R}}(ag)$ from $\to^*_{\mathcal{R}}(bh)$ with regular languages since:

$$\to^*_{\mathcal{R}}(ag) \supseteq \{a^n b^m i \mid n > m\}, \quad \to^*_{\mathcal{R}}(bh) \supseteq \{a^n b^m i \mid n < m\}$$

- represent A, B as arctically $(\{-\infty\} \cup \mathbb{Z}, \max, +)$ weighted automata, with one state each. Encode non-usability by $A(h) = -\infty$, $B(g) = -\infty$.

Example (Ex. 1)

- non-joinability of ag, bh with respect to $\mathcal{R} = \{g \to ag, g \to i, h \to bh, h \to i, i \to abi, ab \to ba, ba \to ab\}$
- algebras $A : s \mapsto \#_a(s) - \#_b(s)$, $B : s \mapsto \#_b(s) - \#_a(s)$,
  - for $s \in \to_{\mathcal{R}}^*(ag) : 1 \le A(s)$ note: $(h \to bh)$ not usable
  - for $s \in \to_{\mathcal{R}}^*(bh) : 1 \le B(s)$ note: $(g \to ag)$ not usable
  - for all $s : A(ag) + B(bh) = 2 \not\le 0 = A(s) + B(s)$
- cannot separate $\to_{\mathcal{R}}^*(ag)$ from $\to_{\mathcal{R}}^*(bh)$ with regular languages since:

$$\to_{\mathcal{R}}^*(ag) \supseteq \{a^n b^m i \mid n > m\}, \quad \to_{\mathcal{R}}^*(bh) \supseteq \{a^n b^m i \mid n < m\}$$

- represent A, B as arctically $(\{-\infty\} \cup \mathbb{Z}, \max, +)$ weighted automata, with one state each.
  Encode non-usability by $A(h) = -\infty$, $B(g) = -\infty$.

Example (Ex. 1)

- non-joinability of ag, bh with respect to $\mathcal{R} = \{g \to ag, g \to i, h \to bh, h \to i, i \to abi, ab \to ba, ba \to ab\}$
- algebras $A : s \mapsto \#_a(s) - \#_b(s)$, $B : s \mapsto \#_b(s) - \#_a(s)$,
  - for $s \in \to_{\mathcal{R}}^*(ag) : 1 \leq A(s)$ note: $(h \to bh)$ not usable
  - for $s \in \to_{\mathcal{R}}^*(bh) : 1 \leq B(s)$ note: $(g \to ag)$ not usable
  - for all $s : A(ag) + B(bh) = 2 \not\leq 0 = A(s) + B(s)$
- cannot separate $\to_{\mathcal{R}}^*(ag)$ from $\to_{\mathcal{R}}^*(bh)$ with regular languages since:

$$\to_{\mathcal{R}}^*(ag) \supseteq \{a^n b^m i \mid n > m\}, \quad \to_{\mathcal{R}}^*(bh) \supseteq \{a^n b^m i \mid n < m\}$$

- represent A, B as arctically $(\{-\infty\} \cup \mathbb{Z}, \max, +)$ weighted automata, with one state each.
  Encode non-usability by $A(h) = -\infty, B(g) = -\infty$.

Example (Ex. 1)

- non-joinability of ag, bh with respect to $\mathcal{R} = \{g \to ag, g \to i, h \to bh, h \to i, i \to abi, ab \to ba, ba \to ab\}$
- algebras $A : s \mapsto \#_a(s) - \#_b(s)$, $B : s \mapsto \#_b(s) - \#_a(s)$,
  - for $s \in \to^*_{\mathcal{R}}(ag) : 1 \le A(s)$ note: $(h \to bh)$ not usable
  - for $s \in \to^*_{\mathcal{R}}(bh) : 1 \le B(s)$ note: $(g \to ag)$ not usable
  - for all $s : A(ag) + B(bh) = 2 \not\le 0 = A(s) + B(s)$
- cannot separate $\to^*_{\mathcal{R}}(ag)$ from $\to^*_{\mathcal{R}}(bh)$ with regular languages since:

$$\to^*_{\mathcal{R}}(ag) \supseteq \{a^n b^m i \mid n > m\}, \quad \to^*_{\mathcal{R}}(bh) \supseteq \{a^n b^m i \mid n < m\}$$

- represent A, B as arctically $(\{-\infty\} \cup \mathbb{Z}, \max, +)$ weighted automata, with one state each.
  Encode non-usability by $A(h) = -\infty, B(g) = -\infty$.

Abstract Non-Joinability Criterion (Thm. 3)

- Let $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ be weakly monotone Σ-algebras such that $\mathcal{R}$ is weakly oriented by both $\mathcal{A}$ and $\mathcal{B}$, $s, t \in \mathcal{T}(\Sigma)$ be ground terms and $\delta : \mathcal{A} \times \mathcal{B} \to \mathcal{C}$ be a pre-homomorphism between weakly monotone Σ-algebras.
  Then s and t are non-joinable provided that for some $c \in \mathcal{C}$,
  1. $\delta([s]^{\mathcal{A}}, [t]^{\mathcal{B}}) \not\leq c$, and
  2. $f^{\mathcal{C}}(c, \ldots, c) \leq c$ for all $f \in \Sigma$.
- application (Ex. 7, compatible tree automata method)
  - A, B: finite automata; weakly oriented: $\mathcal{R}$-closed
  - C: their Cartesian product automaton (for intersection)
  - c: reachable states in C
- next: extend to weighted automata, restrict to strings

Abstract Non-Joinability Criterion (Thm. 3)

- Let $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ be weakly monotone Σ-algebras such that $\mathcal{R}$ is weakly oriented by both $\mathcal{A}$ and $\mathcal{B}$, $s, t \in \mathcal{T}(\Sigma)$ be ground terms and $\delta : \mathcal{A} \times \mathcal{B} \to \mathcal{C}$ be a pre-homomorphism between weakly monotone Σ-algebras.
  Then s and t are non-joinable provided that for some $c \in \mathcal{C}$,
  1. $\delta([s]^{\mathcal{A}}, [t]^{\mathcal{B}}) \not\leq c$, and
  2. $f^{\mathcal{C}}(c, \ldots, c) \leq c$ for all $f \in \Sigma$.
- application (Ex. 7, compatible tree automata method)
  - A, B: finite automata; weakly oriented: $\mathcal{R}$-closed
  - C: their Cartesian product automaton (for intersection)
  - c: reachable states in C
- next: extend to weighted automata, restrict to strings

Abstract Non-Joinability Criterion (Thm. 3)

- Let $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ be weakly monotone Σ-algebras such that $\mathcal{R}$ is weakly oriented by both $\mathcal{A}$ and $\mathcal{B}$, $s, t \in \mathcal{T}(\Sigma)$ be ground terms and $\delta : \mathcal{A} \times \mathcal{B} \to \mathcal{C}$ be a pre-homomorphism between weakly monotone Σ-algebras.
  Then s and t are non-joinable provided that for some $c \in \mathcal{C}$,
  1. $\delta([s]^{\mathcal{A}}, [t]^{\mathcal{B}}) \not\leq c$, and
  2. $f^{\mathcal{C}}(c, \ldots, c) \leq c$ for all $f \in \Sigma$.
- application (Ex. 7, compatible tree automata method)
  - A, B: finite automata; weakly oriented: $\mathcal{R}$-closed
  - C: their Cartesian product automaton (for intersection)
  - c: reachable states in C
- next: extend to weighted automata, restrict to strings

Algebras from Finite Weighted Algebra

- $(S, \leq)$ a totally ordered semi-ring, e.g.,
  natural numbers $(\mathbb{N}, +, \cdot, 0, 1)$,
  arctic integers $(\mathbb{A}, \max, +, -\infty, 0)$, Booleans $(\mathbb{B}, \vee, \wedge, \mathbf{F}, \mathbf{T})$.
- S-weighted tree automaton A over alphabet Σ:
  - set of states Q,
  - family of transition mappings $\mu_k : \Sigma_k \to (Q^k \times Q \to S)$,
  - root weight vector $\nu : Q \to S$.

  The algebra μ_A of this automaton has domain $(Q \to S, \leq)$.
  (Q-indexed vectors of S values, ordered point-wise)
- Kronecker product automaton $A \odot B$ with states $Q_A \times Q_B$,
  $\mu_{A \odot B}(f)((v_A, v_B), (p_A, p_B)) = \mu_A(f)(v_A, p_A) \odot \mu_B(f)(v_B, p_B)$
- current implementation is for strings only,
  as matrix interpretations do not commute with $\odot$

Algebras from Finite Weighted Algebra

- $(S, \leq)$ a totally ordered semi-ring, e.g., natural numbers $(\mathbb{N}, +, \cdot, 0, 1)$, arctic integers $(\mathbb{A}, \max, +, -\infty, 0)$, Booleans $(\mathbb{B}, \vee, \wedge, \mathbf{F}, \mathbf{T})$.
- S-weighted tree automaton A over alphabet Σ:
  - set of states Q,
  - family of transition mappings $\mu_k : \Sigma_k \to (Q^k \times Q \to S)$,
  - root weight vector $\nu : Q \to S$.

  The algebra μ_A of this automaton has domain $(Q \to S, \leq)$. (Q-indexed vectors of S values, ordered point-wise)
- Kronecker product automaton $A \odot B$ with states $Q_A \times Q_B$, $\mu_{A \odot B}(f)((v_A, v_B), (p_A, p_B)) = \mu_A(f)(v_A, p_A) \odot \mu_B(f)(v_B, p_B)$
- current implementation is for strings only, as matrix interpretations do not commute with $\odot$

Algebras from Finite Weighted Algebra

- $(S, \leq)$ a totally ordered semi-ring, e.g., natural numbers $(\mathbb{N}, +, \cdot, 0, 1)$, arctic integers $(\mathbb{A}, \max, +, -\infty, 0)$, Booleans $(\mathbb{B}, \vee, \wedge, \mathbf{F}, \mathbf{T})$.
- S-weighted tree automaton A over alphabet Σ:
  - set of states Q,
  - family of transition mappings $\mu_k : \Sigma_k \to (Q^k \times Q \to S)$,
  - root weight vector $\nu : Q \to S$.

  The algebra μ_A of this automaton has domain $(Q \to S, \leq)$. (Q-indexed vectors of S values, ordered point-wise)
- Kronecker product automaton $A \odot B$ with states $Q_A \times Q_B$, $\mu_{A \odot B}(f)((v_A, v_B), (p_A, p_B)) = \mu_A(f)(v_A, p_A) \odot \mu_B(f)(v_B, p_B)$
- current implementation is for strings only, as matrix interpretations do not commute with $\odot$

Algebras from Finite Weighted Algebra

- $(S, \leq)$ a totally ordered semi-ring, e.g., natural numbers $(\mathbb{N}, +, \cdot, 0, 1)$, arctic integers $(\mathbb{A}, \max, +, -\infty, 0)$, Booleans $(\mathbb{B}, \vee, \wedge, \mathbf{F}, \mathbf{T})$.
- S-weighted tree automaton A over alphabet Σ:
  - set of states Q,
  - family of transition mappings $\mu_k : \Sigma_k \to (Q^k \times Q \to S)$,
  - root weight vector $\nu : Q \to S$.

  The algebra μ_A of this automaton has domain $(Q \to S, \leq)$. (Q-indexed vectors of S values, ordered point-wise)
- Kronecker product automaton $A \odot B$ with states $Q_A \times Q_B$, $\mu_{A \odot B}(f)((v_A, v_B), (p_A, p_B)) = \mu_A(f)(v_A, p_A) \odot \mu_B(f)(v_B, p_B)$
- current implementation is for strings only, as matrix interpretations do not commute with $\odot$

Implementation: Noko Leipzig

- for proving Nonkonfluenz (and it rhymes with a TV series)
- Noko Leipzig is part of Matchbox `https://gitlab.imn.htwk-leipzig.de/waldmann/pure-matchbox`
- core functionality: prove non-joinability
  - input: SRS $\mathcal{R}$ over Σ; $s, t \in \Sigma^*$; $d, b \in \mathbb{N}$.
  - output (if successful): arctically weighted automata A, B with d states, weights represented by b bits, and arctic vector $c \in (Q_A \times Q_B \to \mathbb{A})$, that fulfil the conditions of Theorem 3
- transform to a Boolean satisfiability problem with the Ersatz library (Kmett 201?), solve with Minisat (Sörensen 200?)
- performance in CoCo 2019 (for SRS): 6 unique NO answers, two (Cops 1034, 1131) using automata.

Implementation: Noko Leipzig

- for proving Nonkonfluenz (and it rhymes with a TV series)
- Noko Leipzig is part of Matchbox `https://gitlab.imn.htwk-leipzig.de/waldmann/pure-matchbox`
- core functionality: prove non-joinability
  - input: SRS $\mathcal{R}$ over Σ; $s, t \in \Sigma^*$; $d, b \in \mathbb{N}$.
  - output (if successful): arctically weighted automata $\mathcal{A}, \mathcal{B}$ with d states, weights represented by b bits, and arctic vector $c \in (Q_A \times Q_B \to \mathbb{A})$, that fulfil the conditions of Theorem 3
- transform to a Boolean satisfiability problem with the Ersatz library (Kmett 201?), solve with Minisat (Sörensen 200?)
- performance in CoCo 2019 (for SRS): 6 unique NO answers, two (Cops 1034, 1131) using automata.

Implementation: Noko Leipzig

- for proving Nonkonfluenz (and it rhymes with a TV series)
- Noko Leipzig is part of Matchbox `https://gitlab.imn.htwk-leipzig.de/waldmann/pure-matchbox`
- core functionality: prove non-joinability
  - input: SRS $\mathcal{R}$ over Σ; $s, t \in \Sigma^*$; $d, b \in \mathbb{N}$.
  - output (if successful): arctically weighted automata A, B with d states, weights represented by b bits, and arctic vector $c \in (Q_A \times Q_B \to \mathbb{A})$, that fulfil the conditions of Theorem 3
- transform to a Boolean satisfiability problem with the Ersatz library (Kmett 201?), solve with Minisat (Sörensen 200?)
- performance in CoCo 2019 (for SRS): 6 unique NO answers, two (Cops 1034, 1131) using automata.

Implementation: Noko Leipzig

- for proving Nonkonfluenz (and it rhymes with a TV series)
- Noko Leipzig is part of Matchbox `https://gitlab.imn.htwk-leipzig.de/waldmann/pure-matchbox`
- core functionality: prove non-joinability
  - input: SRS $\mathcal{R}$ over Σ; $s, t \in \Sigma^*$; $d, b \in \mathbb{N}$.
  - output (if successful): arctically weighted automata A, B with d states, weights represented by b bits, and arctic vector $c \in (Q_A \times Q_B \to \mathbb{A})$, that fulfil the conditions of Theorem 3
- transform to a Boolean satisfiability problem with the Ersatz library (Kmett 201?), solve with Minisat (Sörensen 200?)
- performance in CoCo 2019 (for SRS): 6 unique NO answers, two (Cops 1034, 1131) using automata.

Implementation: Noko Leipzig

- for proving Nonkonfluenz (and it rhymes with a TV series)
- Noko Leipzig is part of Matchbox `https://gitlab.imn.htwk-leipzig.de/waldmann/pure-matchbox`
- core functionality: prove non-joinability
  - input: SRS $\mathcal{R}$ over Σ; $s, t \in \Sigma^*$; $d, b \in \mathbb{N}$.
  - output (if successful): arctically weighted automata A, B with d states, weights represented by b bits, and arctic vector $c \in (Q_A \times Q_B \to \mathbb{A})$, that fulfil the conditions of Theorem 3
- transform to a Boolean satisfiability problem with the Ersatz library (Kmett 201?), solve with Minisat (Sörensen 200?)
- performance in CoCo 2019 (for SRS): 6 unique NO answers, two (Cops 1034, 1131) using automata.

Implementation: Noko Leipzig

- for proving Nonkonfluenz (and it rhymes with a TV series)
- Noko Leipzig is part of Matchbox `https://gitlab.imn.htwk-leipzig.de/waldmann/pure-matchbox`
- core functionality: prove non-joinability
  - input: SRS $\mathcal{R}$ over Σ; $s, t \in \Sigma^*$; $d, b \in \mathbb{N}$.
  - output (if successful): arctically weighted automata A, B with d states, weights represented by b bits, and arctic vector $c \in (Q_A \times Q_B \to \mathbb{A})$, that fulfil the conditions of Theorem 3
- transform to a Boolean satisfiability problem with the Ersatz library (Kmett 201?), solve with Minisat (Sörensen 200?)
- performance in CoCo 2019 (for SRS): 6 unique NO answers, two (Cops 1034, 1131) using automata.

Implementation: Noko Leipzig

- for proving Nonkonfluenz (and it rhymes with a TV series)
- Noko Leipzig is part of Matchbox `https://gitlab.imn.htwk-leipzig.de/waldmann/pure-matchbox`
- core functionality: prove non-joinability
  - input: SRS $\mathcal{R}$ over Σ; $s, t \in \Sigma^*$; $d, b \in \mathbb{N}$.
  - output (if successful): arctically weighted automata A, B with d states, weights represented by b bits, and arctic vector $c \in (Q_A \times Q_B \to \mathbb{A})$, that fulfil the conditions of Theorem 3
- transform to a Boolean satisfiability problem with the Ersatz library (Kmett 201?), solve with Minisat (Sörensen 200?)
- performance in CoCo 2019 (for SRS): 6 unique NO answers, two (Cops 1034, 1131) using automata.

An Example (21538)

- rules $R = R_1 \cup R_2$ where
  $R_1 = \{ba \to cab, ca \to aba\}, R_2 = \{da \to bdd, dc \to cbb\}$
- peak $s = cbba \leftarrow dca \to daba = t$
- $A =$

, $B =$

- $A(s) = -1, B(t) = 3, \forall x : A(x) \cdot B(x) \in \{-\infty, 0\}$
- $A \odot B$ is (weakly increasing and) not constant
  (if last d vanishes, it jumps from $-\infty$ to 0)
- notes: A is constant on R_1. R_2 is not usable for s.
- cannot be separated by regular languages? cannot be separated by arctic automata with just one state?

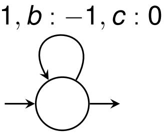
An Example (21538)

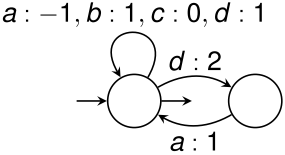
- rules $R = R_1 \cup R_2$ where
  $R_1 = \{ba \to cab, ca \to aba\}, R_2 = \{da \to bdd, dc \to cbb\}$
- peak $s = cbba \leftarrow dca \to daba = t$
- $A =$

, $B =$

- $A(s) = -1, B(t) = 3, \forall x : A(x) \cdot B(x) \in \{-\infty, 0\}$
- $A \odot B$ is (weakly increasing and) not constant
  (if last d vanishes, it jumps from $-\infty$ to 0)
- notes: A is constant on R_1. R_2 is not usable for s.
- cannot be separated by regular languages? cannot be separated by arctic automata with just one state?

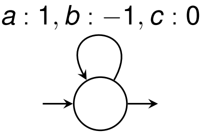
An Example (21538)

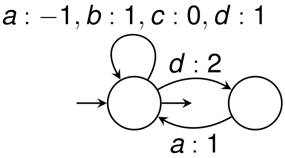
- rules $R = R_1 \cup R_2$ where
  $R_1 = \{ba \to cab, ca \to aba\}, R_2 = \{da \to bdd, dc \to cbb\}$
- peak $s = cbba \leftarrow dca \to daba = t$
- $A =$ 

  , $B =$ 

- $A(s) = -1, B(t) = 3, \forall x : A(x) \cdot B(x) \in \{-\infty, 0\}$
- $A \odot B$ is (weakly increasing and) not constant
  (if last d vanishes, it jumps from $-\infty$ to 0)
- notes: A is constant on R_1. R_2 is not usable for s.
- cannot be separated by regular languages? cannot be separated by arctic automata with just one state?

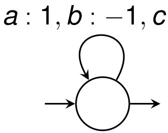
An Example (21538)

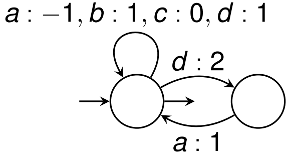
- rules $R = R_1 \cup R_2$ where $R_1 = \{ba \to cab, ca \to aba\}, R_2 = \{da \to bdd, dc \to cbb\}$
- peak $s = cbba \leftarrow dca \to daba = t$
- $A =$ 

, $B =$ 

- $A(s) = -1, B(t) = 3, \forall x : A(x) \cdot B(x) \in \{-\infty, 0\}$
- $A \odot B$ is (weakly increasing and) not constant (if last d vanishes, it jumps from $-\infty$ to 0)
- notes: A is constant on R_1. R_2 is not usable for s.
- cannot be separated by regular languages? cannot be separated by arctic automata with just one state?

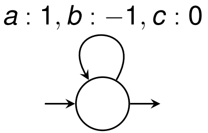
An Example (21538)

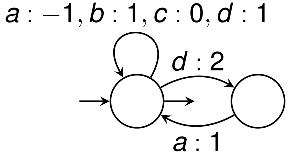
- rules $R = R_1 \cup R_2$ where
  $R_1 = \{ba \to cab, ca \to aba\}, R_2 = \{da \to bdd, dc \to cbb\}$
- peak $s = cbba \leftarrow dca \to daba = t$
- $A =$ 

, $B =$ 

- $A(s) = -1, B(t) = 3, \forall x : A(x) \cdot B(x) \in \{-\infty, 0\}$
- $A \odot B$ is (weakly increasing and) not constant
  (if last d vanishes, it jumps from $-\infty$ to 0)
- notes: A is constant on R_1. R_2 is not usable for s.
- cannot be separated by regular languages? cannot be separated by arctic automata with just one state?

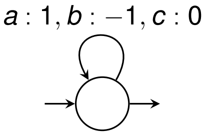
An Example (21538)

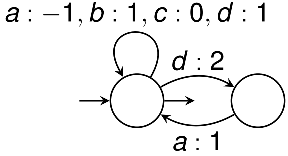
- rules $R = R_1 \cup R_2$ where
  $R_1 = \{ba \to cab, ca \to aba\}, R_2 = \{da \to bdd, dc \to cbb\}$
- peak $s = cbba \leftarrow dca \to daba = t$
- $A =$ 

  , $B =$ 

- $A(s) = -1, B(t) = 3, \forall x : A(x) \cdot B(x) \in \{-\infty, 0\}$
- $A \odot B$ is (weakly increasing and) not constant
  (if last d vanishes, it jumps from $-\infty$ to 0)
- notes: A is constant on R_1. R_2 is not usable for s.
- cannot be separated by regular languages? cannot be separated by arctic automata with just one state?

An Example (21538)

- rules $R = R_1 \cup R_2$ where
  $R_1 = \{ba \to cab, ca \to aba\}, R_2 = \{da \to bdd, dc \to cbb\}$
- peak $s = cbba \leftarrow dca \to daba = t$
- $A =$ 

  , $B =$ 

- $A(s) = -1, B(t) = 3, \forall x : A(x) \cdot B(x) \in \{-\infty, 0\}$
- $A \odot B$ is (weakly increasing and) not constant
  (if last d vanishes, it jumps from $-\infty$ to 0)
- notes: A is constant on R_1. R_2 is not usable for s.
- cannot be separated by regular languages? cannot be separated by arctic automata with just one state?

Another Example (14848)

- rules $\{dc \to dbb, cb \to bcc, db \to dcd, bc \to bcb\}$.
- peak $s = dcdc \leftarrow dbc \to dbcb = t$.
- $A =$

, $B =$

-

$A \odot B =$

Another Example (14848)

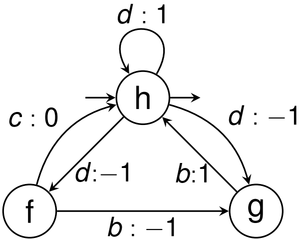
- rules $\{dc \to dbb, cb \to bcc, db \to dcd, bc \to bcb\}$.
- peak $s = dcdc \leftarrow dbc \to dbcb = t$.
- $A =$

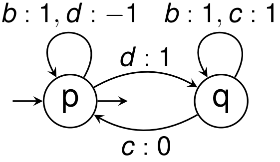
, $B =$

-

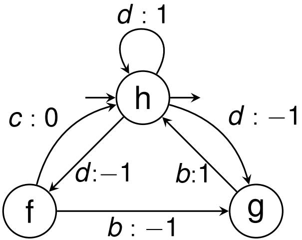
$A \odot B =$

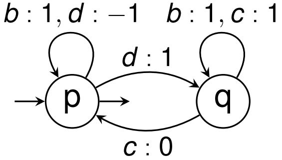
Another Example (14848)

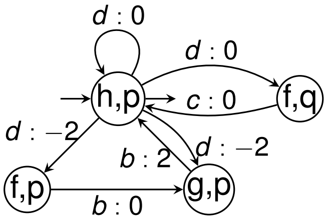
- rules $\{dc \to dbb, cb \to bcc, db \to dcd, bc \to bcb\}$.
- peak $s = dcdc \leftarrow dbc \to dbcb = t$.
- $A =$ 

, $B =$ 

Another Example (14848)

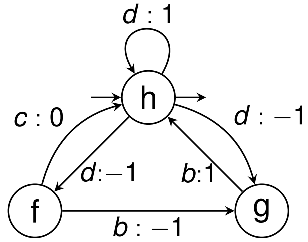
- rules $\{dc \to dbb, cb \to bcc, db \to dcd, bc \to bcb\}$.
- peak $s = dcdc \leftarrow dbc \to dbcb = t$.
- $A =$ 

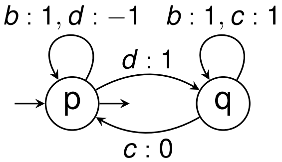
, $B =$ 

- $A \odot B =$ 

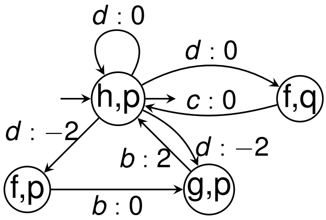
- $A(s) = -2, B(t) = 3$
- $\forall x : A(x) \cdot B(x) \leq 0$

Another Example (14848)

- rules $\{dc \to dbb, cb \to bcc, db \to dcd, bc \to bcb\}$.
- peak $s = dcdc \leftarrow dbc \to dbcb = t$.
- $A =$ 

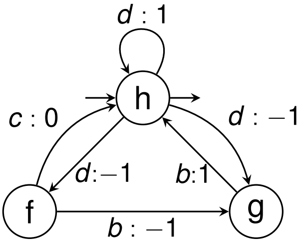
, $B =$ 

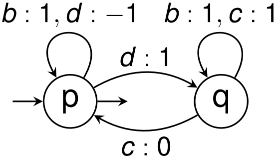
-

$A \odot B =$ 

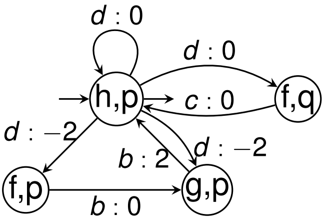
- $A(s) = -2, B(t) = 3$
- $\forall x : A(x) \cdot B(x) \leq 0$

Another Example (14848)

- rules $\{dc \to dbb, cb \to bcc, db \to dcd, bc \to bcb\}$.
- peak $s = dcdc \leftarrow dbc \to dbcb = t$.
- $A =$ 

$, B =$ 

-

$A \odot B =$ 

- $A(s) = -2, B(t) = 3$
- $\forall x : A(x) \cdot B(x) \leq 0$

Related Work, Discussion

- contains “compatible tree automata method” (Zankl et al. 2011) as special case
- is certainly related to *Disproving Confluence by… Ordering* (Aoto, 2013), …but how exactly?
  Both show that $\delta([s]^{\mathcal{A}}, [t]^{\mathcal{B}}) \not\leq \delta([u]^{\mathcal{A}}, [u]^{\mathcal{B}})$ for all u.
  Aoto: $\mathcal{B}$ as *opposite* of $\mathcal{A}$, check $[s]^{\mathcal{A}} \not\leq [t]^{\mathcal{A}}$, rules out that

  $$([s]^{\mathcal{A}}, [t]^{\mathcal{B}}) \leq ([u]^{\mathcal{A}}, [u]^{\mathcal{B}}) \iff [s]^{\mathcal{A}} \leq [u]^{\mathcal{A}} \leq [t]^{\mathcal{A}}$$

  We establish upper bound on $\delta([u]^{\mathcal{A}}, [u]^{\mathcal{B}})$ by induction on u.
- implementation (constraint solving) is expensive — too much for tight CoCo settings
- “killer examples” (no Boolean automaton at all, not 1-state arctic automaton) are few, and far between

Related Work, Discussion

- contains "compatible tree automata method" (Zankl et al. 2011) as special case
- is certainly related to *Disproving Confluence by... Ordering* (Aoto, 2013), ... but how exactly?
  Both show that $\delta([s]^{\mathcal{A}}, [t]^{\mathcal{B}}) \not\leq \delta([u]^{\mathcal{A}}, [u]^{\mathcal{B}})$ for all u.
  Aoto: $\mathcal{B}$ as *opposite* of $\mathcal{A}$, check $[s]^{\mathcal{A}} \not\leq [t]^{\mathcal{A}}$, rules out that

  $$([s]^{\mathcal{A}}, [t]^{\mathcal{B}}) \leq ([u]^{\mathcal{A}}, [u]^{\mathcal{B}}) \iff [s]^{\mathcal{A}} \leq [u]^{\mathcal{A}} \leq [t]^{\mathcal{A}}$$

  We establish upper bound on $\delta([u]^{\mathcal{A}}, [u]^{\mathcal{B}})$ by induction on u.
- implementation (constraint solving) is expensive — too much for tight CoCo settings
- "killer examples" (no Boolean automaton at all, not 1-state arctic automaton) are few, and far between