Language Inclusion Algorithms as Complete Abstract Interpretations

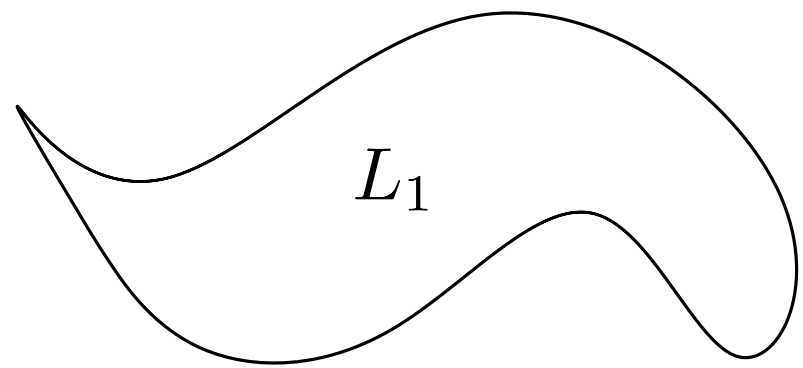
Pierre Ganty
Pedro Valero
IMDEA Software Institute

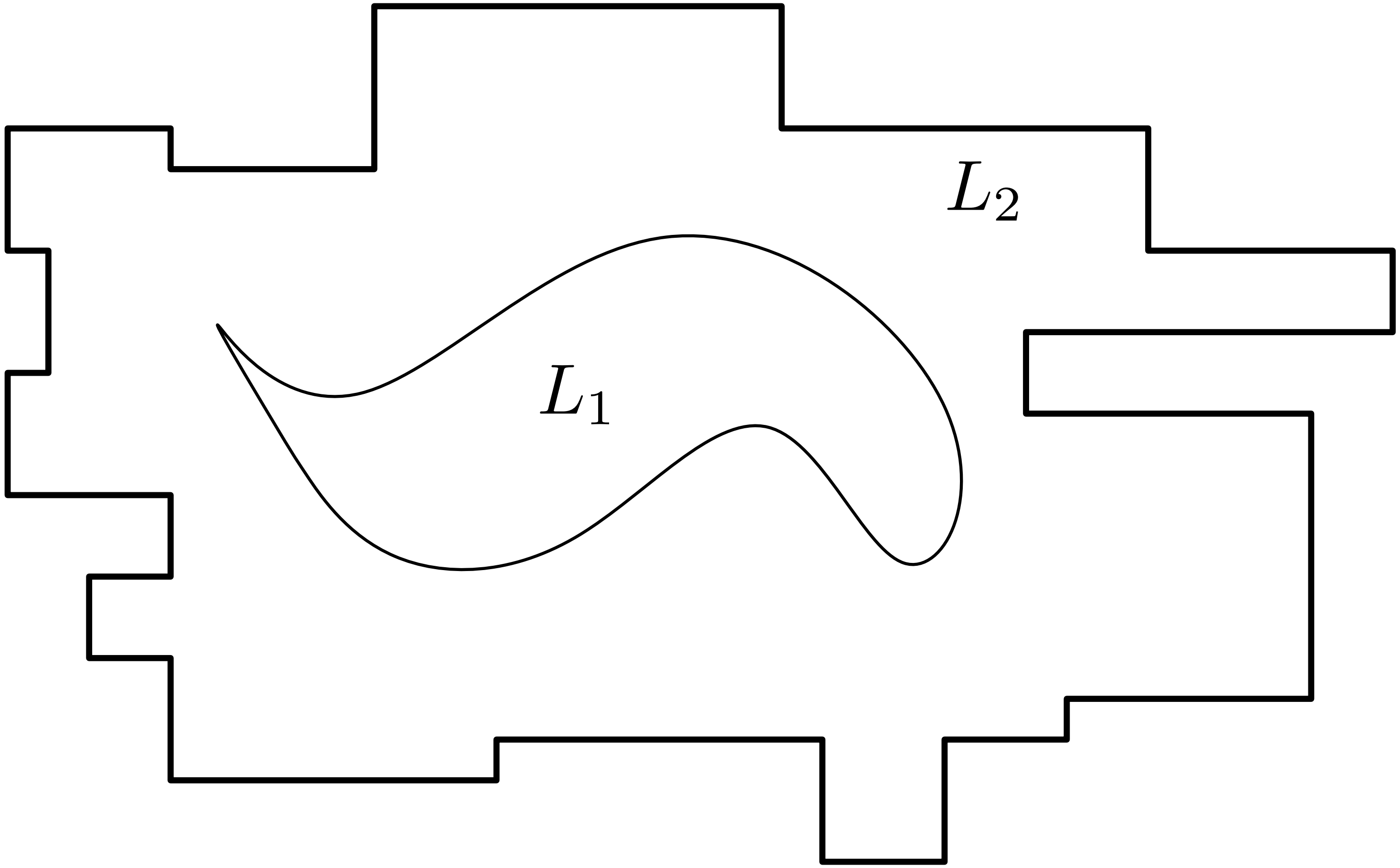
Francesco Ranzato
University of Padova

Abstract Interpretation from Büchi Automata

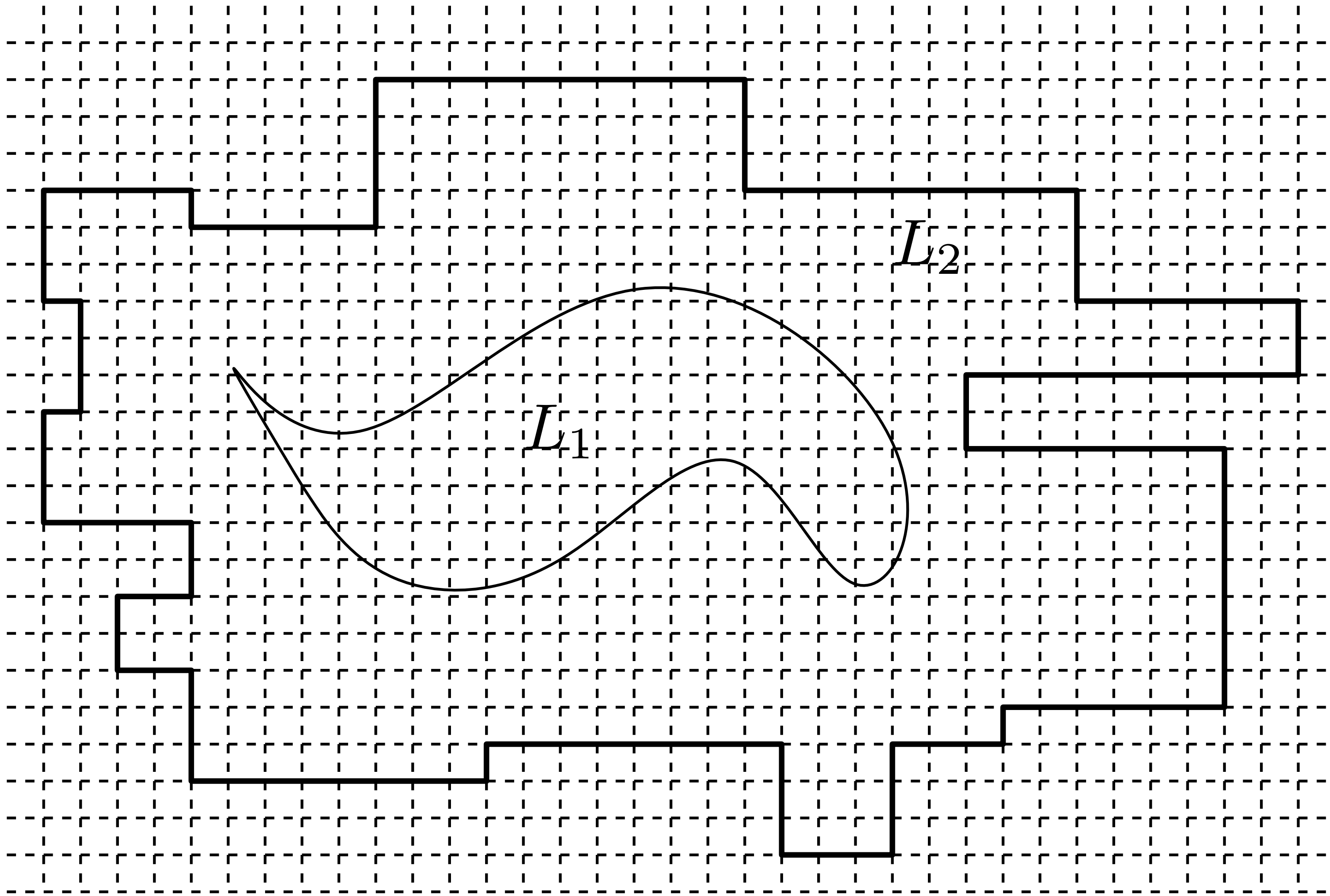
Martin Hofmann
LMU Munich, Germany
martin.hofmann@ifi.lmu.de

Wei Chen
University of Edinburgh, UK
wchen2@inf.ed.ac.uk

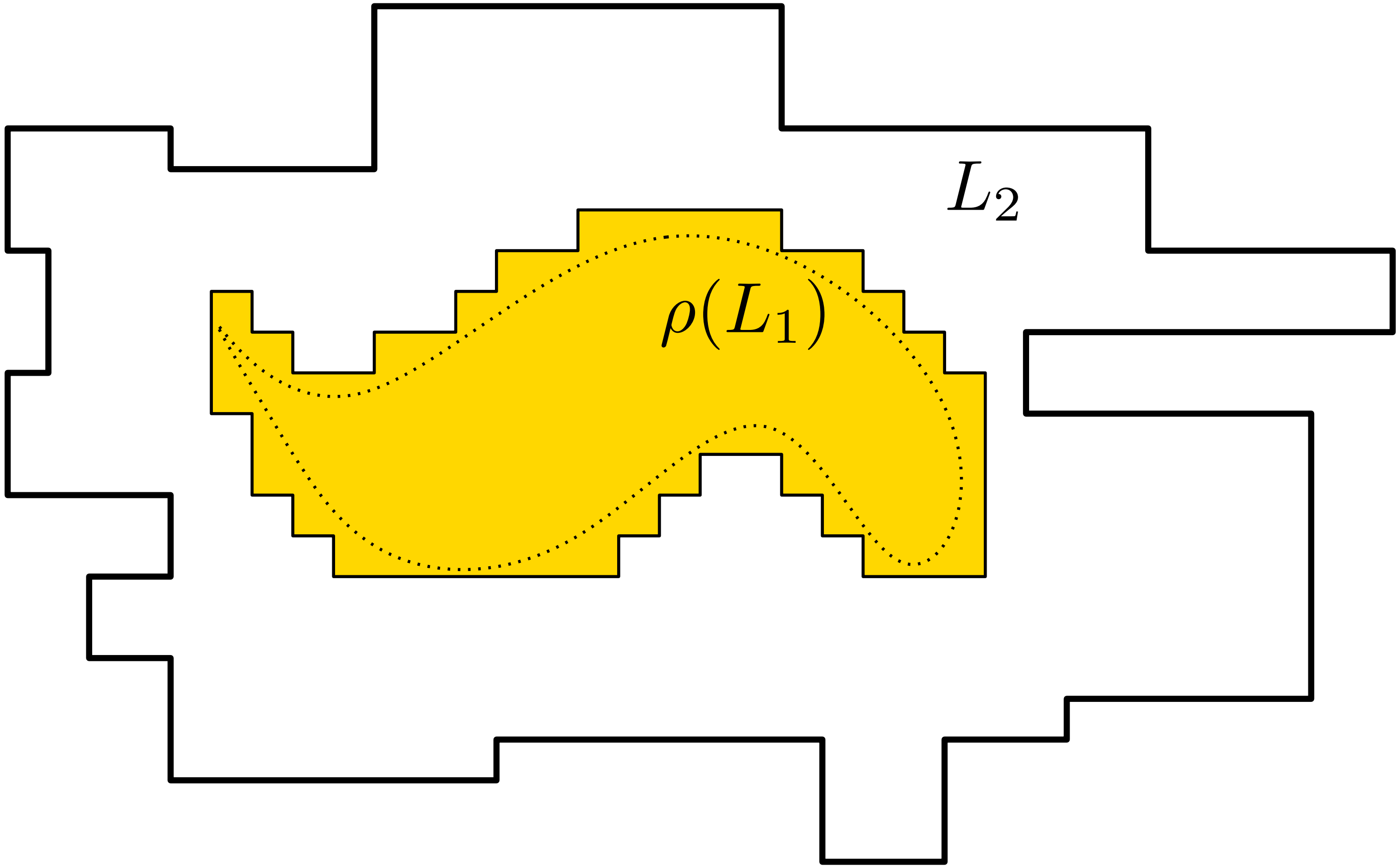

L_1

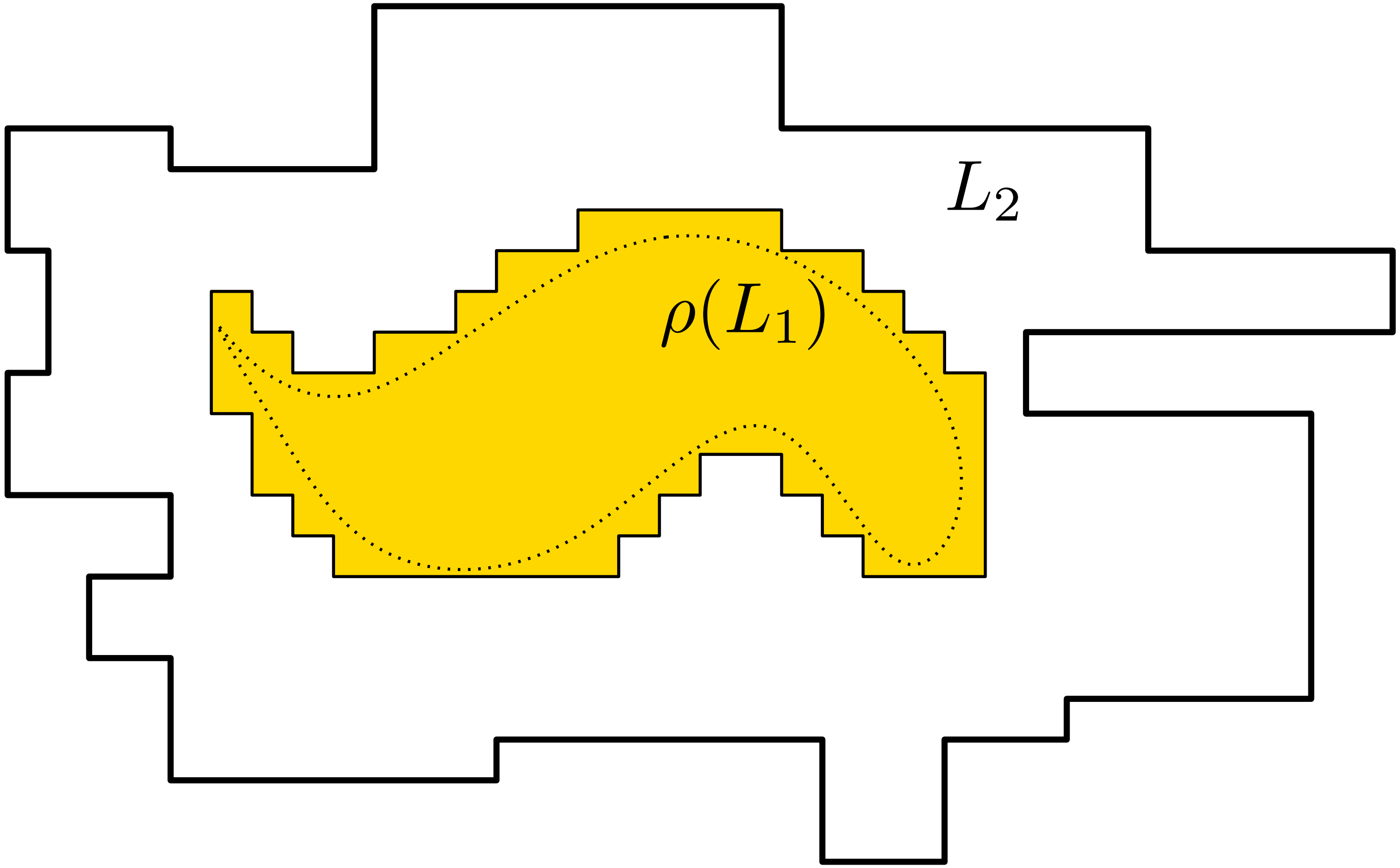


L_2
L_1

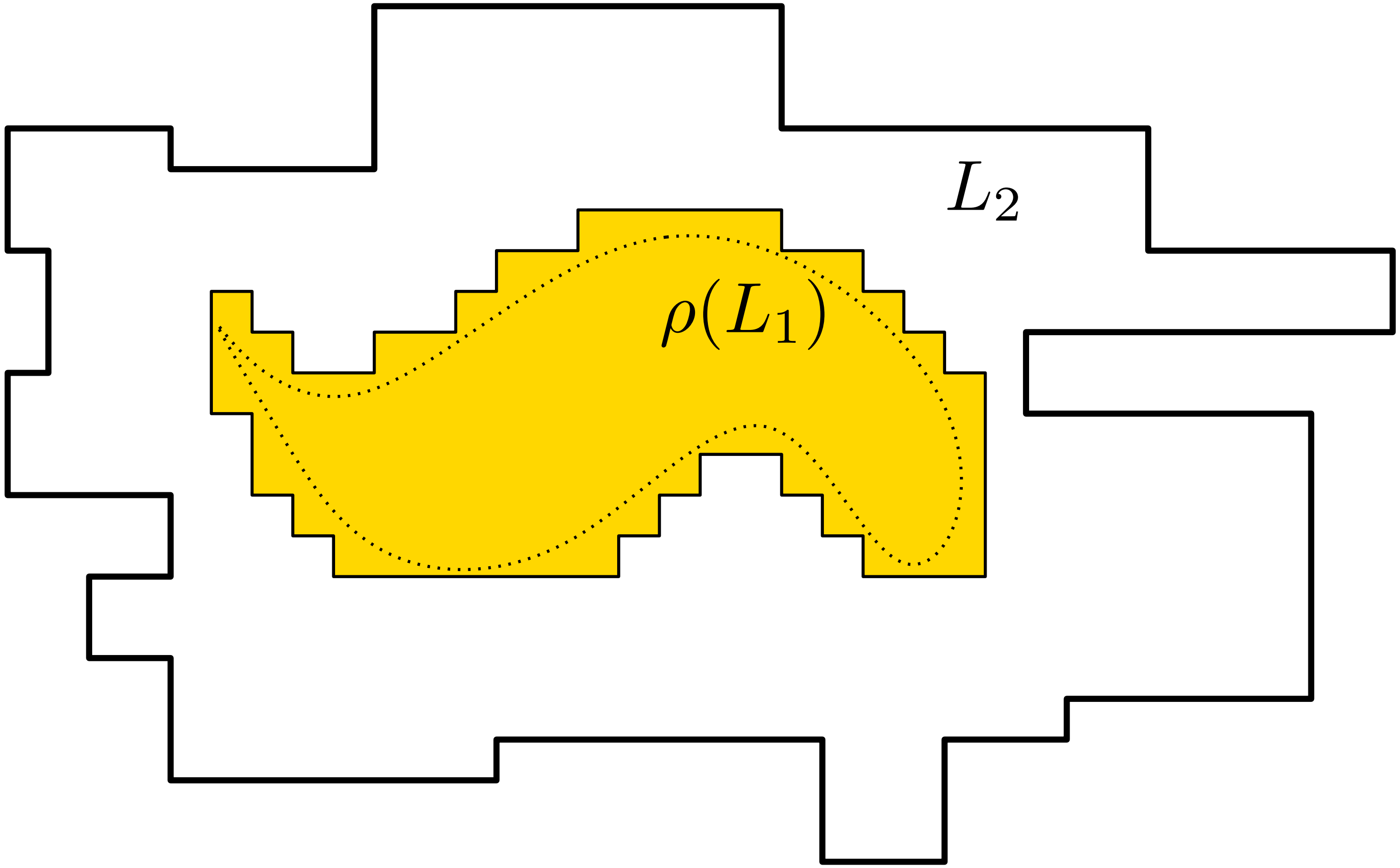


L_2
L_1


L_2
$\rho(L_1)$

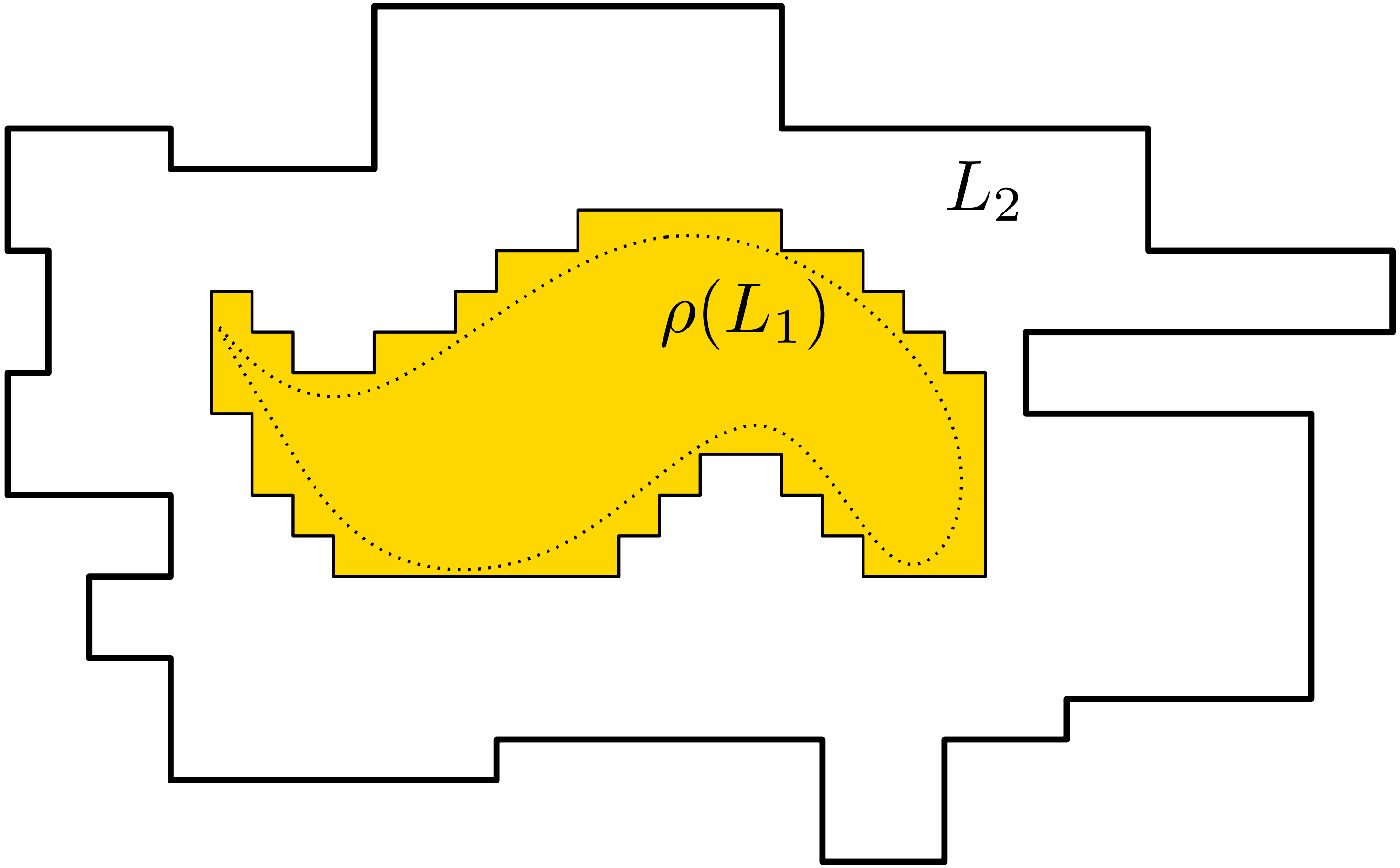


- ρ is extensive: $L \subseteq \rho(L)$
- ρ is monotone: $L \subseteq L'$ implies $\rho(L) \subseteq \rho(L')$
- ρ is idempotent: $\rho(\rho(L)) = \rho(L)$



Assume $\rho(L_2) = L_2$, then

$$L_1 \subseteq L_2 \text{ iff } \rho(L_1) \subseteq L_2$$



Assume $\rho(L_2) = L_2$, then

$$L_1 \subseteq L_2 \text{ iff } \rho(L_1) \subseteq L_2$$

$$\rho(L_2) = L_2$$

L_1 as a fixed point

$$X \to aX \mid \epsilon$$

L_1 as a fixed point

$$X \to aX \mid \epsilon$$

$$\mathrm{lfp}\, \lambda X.\, (aX \cup \{\epsilon\})$$

L_1 as a fixed point

$$X \to aX \mid \epsilon$$

$$\mathrm{lfp}\, \lambda X.\, (aX \cup \{\epsilon\})$$

$\varnothing$

L_1 as a fixed point

$$X \to aX \mid \epsilon$$

$$\mathrm{lfp}\, \lambda X.\, (aX \cup \{\epsilon\})$$

$\varnothing$ $\{\epsilon\}$

L_1 as a fixed point

$$X \to aX \mid \epsilon$$

$$\operatorname{lfp} \lambda X.\, (aX \cup \{\epsilon\})$$

$\varnothing$ $\{\epsilon\}$ $\{\epsilon, a\}$

L_1 as a fixed point

$$X \to aX \mid \epsilon$$

$$\mathrm{lfp}\, \lambda X.\, (aX \cup \{\epsilon\})$$

$$\varnothing \qquad \{\epsilon\} \qquad \{\epsilon, a\} \qquad \{\epsilon, a, a^2\}$$

L_1 as a fixed point

$$X \to aX \mid \epsilon$$

$$\mathrm{lfp}\, \lambda X.\, (aX \cup \{\epsilon\})$$

$$\varnothing \qquad \{\epsilon\} \qquad \{\epsilon, a\} \qquad \{\epsilon, a, a^2\} \bullet \quad \bullet \quad \bullet$$

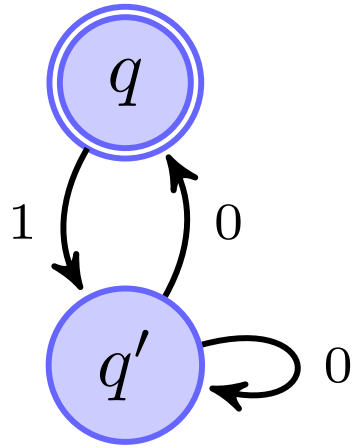
L_1 as a fixed point

$$X \to aX \mid \epsilon$$

$$\operatorname{lfp} \lambda X.\, (aX \cup \{\epsilon\})$$

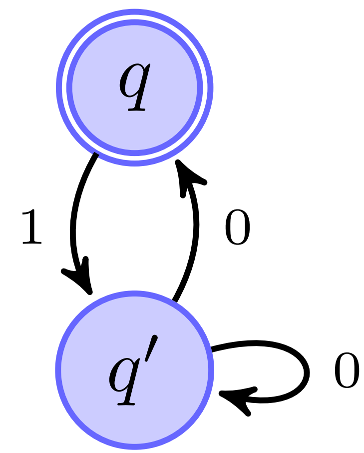
$$\varnothing \qquad \{\epsilon\} \qquad \{\epsilon, a\} \qquad \{\epsilon, a, a^2\} \bullet \quad \bullet \quad \bullet \{a^n \mid n \geq 0\} = a^*$$

L_1 as a fixed point (cont'd)



$$\mathrm{lfp}\left(\lambda\begin{pmatrix} X_q \\ X_{q'} \end{pmatrix}.\begin{pmatrix} \{\epsilon\} & \cup & 1X_{q'} \\ \varnothing & \cup & 0X_q \cup 0X_{q'} \end{pmatrix}\right)$$

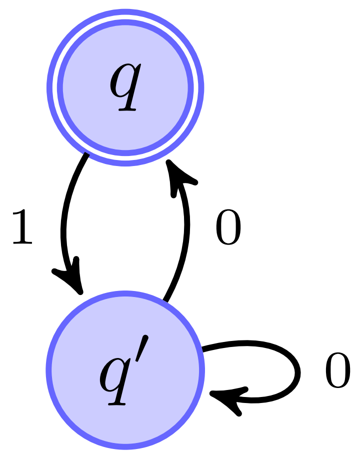
L_1 as a fixed point (cont'd)



$$\mathrm{lfp}\left(\lambda \begin{pmatrix} X_q \\ X_{q'} \end{pmatrix} . \begin{pmatrix} \{\epsilon\} & \cup & 1X_{q'} \\ \varnothing & \cup & 0X_q \cup 0X_{q'} \end{pmatrix}\right)$$

$$\begin{matrix} X_q \\ X_{q'} \end{matrix} \begin{pmatrix} \varnothing \\ \varnothing \end{pmatrix}$$

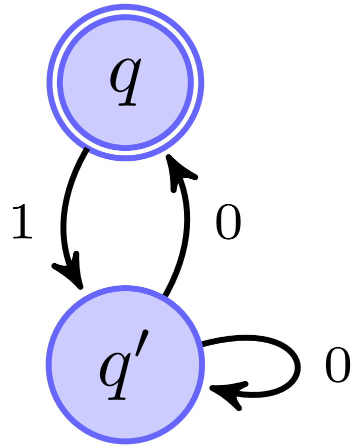
L_1 as a fixed point (cont'd)



$$\mathrm{lfp}\left(\lambda\begin{pmatrix} X_q \\ X_{q'} \end{pmatrix}.\begin{pmatrix} \{\epsilon\} & \cup & 1X_{q'} \\ \varnothing & \cup & 0X_q \cup 0X_{q'} \end{pmatrix}\right)$$

$$\begin{matrix} X_q \\ X_{q'} \end{matrix} \quad \begin{pmatrix} \varnothing \\ \varnothing \end{pmatrix} \quad \begin{pmatrix} \{\varepsilon\} \\ \varnothing \end{pmatrix}$$

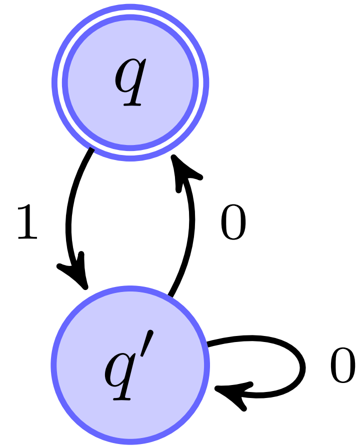
L_1 as a fixed point (cont'd)



$$\mathrm{lfp}\left(\lambda\begin{pmatrix} X_q \\ X_{q'} \end{pmatrix}.\begin{pmatrix} \{\epsilon\} & \cup & 1X_{q'} \\ \varnothing & \cup & 0X_q \cup 0X_{q'} \end{pmatrix}\right)$$

$$\begin{matrix} X_q \\ X_{q'} \end{matrix}\begin{pmatrix} \varnothing \\ \varnothing \end{pmatrix}\quad\begin{pmatrix} \{\varepsilon\} \\ \varnothing \end{pmatrix}\quad\begin{pmatrix} \{\varepsilon\} \\ \{0\} \end{pmatrix}$$

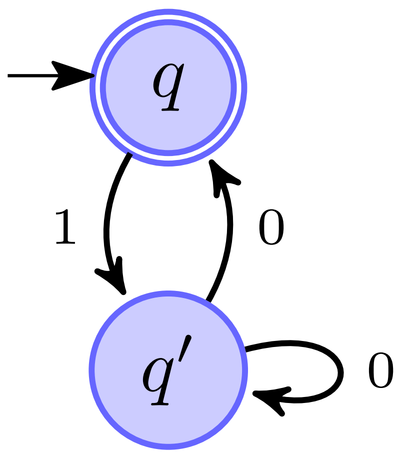
L_1 as a fixed point (cont'd)



$$\mathrm{lfp}\left(\lambda\begin{pmatrix} X_q \\ X_{q'} \end{pmatrix}.\begin{pmatrix} \{\epsilon\} & \cup & 1X_{q'} \\ \varnothing & \cup & 0X_q \cup 0X_{q'} \end{pmatrix}\right)$$

$$\begin{matrix} X_q \\ X_{q'} \end{matrix}\begin{pmatrix} \varnothing \\ \varnothing \end{pmatrix} \quad \begin{pmatrix} \{\varepsilon\} \\ \varnothing \end{pmatrix} \quad \begin{pmatrix} \{\varepsilon\} \\ \{0\} \end{pmatrix} \quad \cdots \quad \begin{matrix} L_q \\ L_{q'} \end{matrix}\begin{pmatrix} (10^+)^* \\ 0^+(10^+)^* \end{pmatrix}$$

L_1 as a fixed point (cont'd)

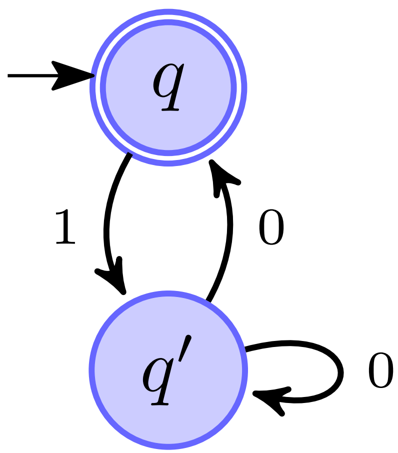


$$\mathrm{lfp}\left(\lambda\begin{pmatrix} X_q \\ X_{q'} \end{pmatrix}.\begin{pmatrix} \{\epsilon\} & \cup & 1X_{q'} \\ \varnothing & \cup & 0X_q \cup 0X_{q'} \end{pmatrix}\right)$$

$$\begin{matrix} X_q \\ X_{q'} \end{matrix} \begin{pmatrix} \varnothing \\ \varnothing \end{pmatrix} \quad \begin{pmatrix} \{\varepsilon\} \\ \varnothing \end{pmatrix} \quad \begin{pmatrix} \{\varepsilon\} \\ \{0\} \end{pmatrix} \quad \cdots \quad \begin{matrix} L_q \\ L_{q'} \end{matrix} \begin{pmatrix} (10^+)^* \\ 0^+(10^+)^* \end{pmatrix}$$

$$L_1 = L_q = (10^+)^* = \langle \mathrm{lfp}\,\lambda X.\, b \cup Fn(X) \rangle_q$$

L_1 as a fixed point (cont'd)

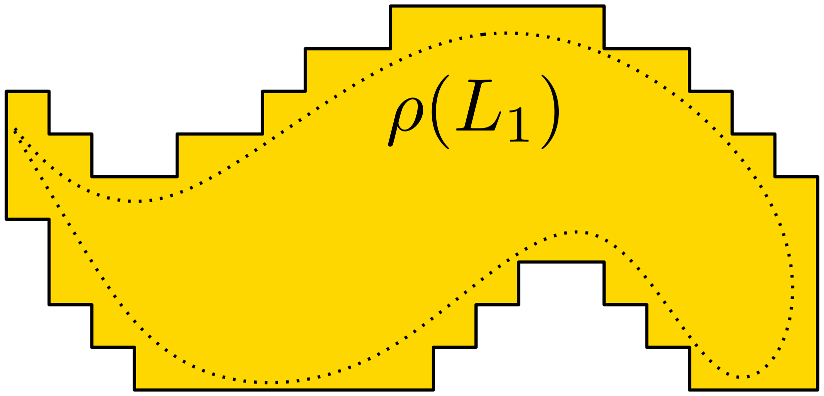


b $\quad$ Fn

$$\mathrm{lfp}\left(\lambda \begin{pmatrix} X_q \\ X_{q'} \end{pmatrix} . \begin{pmatrix} \{\epsilon\} & \cup & 1X_{q'} \\ \emptyset & \cup & 0X_q \cup 0X_{q'} \end{pmatrix}\right)$$

$$\begin{matrix} X_q \\ X_{q'} \end{matrix} \begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix} \quad \begin{pmatrix} \{\varepsilon\} \\ \emptyset \end{pmatrix} \quad \begin{pmatrix} \{\varepsilon\} \\ \{0\} \end{pmatrix} \quad \bullet \bullet \bullet \quad \begin{matrix} L_q \\ L_{q'} \end{matrix} \begin{pmatrix} (10^+)^* \\ 0^+(10^+)^* \end{pmatrix}$$

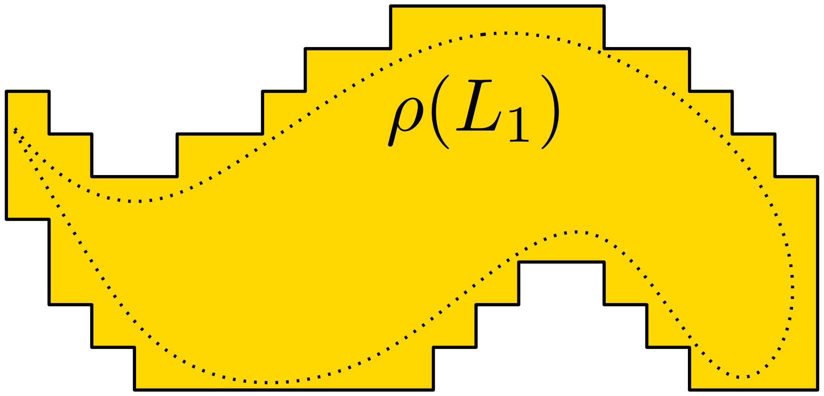
$$L_1 = L_q = (10^+)^* = \langle \mathrm{lfp}\, \lambda X.\, b \cup Fn(X) \rangle_q$$



$$\rho(L_1) = \rho\big(\mathrm{lfp}\,\lambda X.\, b \cup Fn(X)\big)$$

if ρ is complete for Fn (i.e. $\rho\ Fn\ \rho = \rho\ Fn$) then

$$\rho(L_1) = \mathrm{lfp}\,\lambda X.\, \rho\big(b \cup Fn(X)\big)$$



$$\rho(L_1) = \rho\big(\mathrm{lfp}\,\lambda X.\, b \cup Fn(X)\big)$$

if ρ is complete for Fn (i.e. $\rho\ Fn\ \rho = \rho\ Fn$) then

$$\rho(L_1) = \mathrm{lfp}\,\lambda X.\, \rho\big(b \cup Fn(X)\big)$$

ρ is complete for Fn

Wanted

$\rho(L_2) = L_2$

ρ is complete for Fn

Quasiorder induced ρ

Given a quasiorder $\leqslant$ on words, define

$$\rho_{\leqslant}(X) = \{y \mid \exists x \in X \colon x \leqslant y\}$$

Quasiorder induced ρ

Given a quasiorder $\leqslant$ on words, define

$$\rho_{\leqslant}(X) = \{y \mid \exists x \in X : x \leqslant y\}$$

$\leqslant \cap\, (L_2 \times \overline{L_2}) = \varnothing$

$x \leqslant y$ implies $a\,x \leqslant a\,y$ for all a

Quasiorder induced ρ

Given a quasiorder $\leqslant$ on words, define

$$\rho_{\leqslant}(X) = \{y \mid \exists x \in X : x \leqslant y\}$$

$\leqslant \cap\, (L_2 \times \overline{L_2}) = \varnothing$

$x \leqslant y$ implies $a\,x \leqslant a\,y$ for all a

$$\Leftrightarrow$$

$\rho_{\leqslant}(L_2) = L_2$

$\rho_{\leqslant}$ is complete for $\lambda X.\, aX$ for all a

Quasiorder induced ρ

Given a quasiorder $\leqslant$ on words, define

$$\rho_{\leqslant}(X) = \{y \mid \exists x \in X : x \leqslant y\}$$

$\leqslant$ is a left L_2 consistent qo

$\leqslant \cap (L_2 \times \overline{L_2}) = \varnothing$
$x \leqslant y$ implies $a\,x \leqslant a\,y$ for all a

$$\Leftrightarrow$$

$\rho_{\leqslant}(L_2) = L_2$
$\rho_{\leqslant}$ is complete for $\lambda X.\, aX$ for all a

Example of quasiorder

Nerode left quasiorder relative to L_2 [Varricchio,deLuca'94]

$$x \leqslant^{l}_{L_2} y \Leftrightarrow L_2\, x^{-1} \subseteq L_2\, y^{-1}$$

Example of quasiorder

Nerode left quasiorder relative to L

$L\,x^{-1} = \{w \mid wx \in L\}$

$$x \leqslant^{l}_{L_2} y \Leftrightarrow L_2\,x^{-1} \subseteq L_2\,y^{-1}$$

Example of quasiorder

Nerode left quasiorder relative to L

$L\,x^{-1} = \{w \mid wx \in L\}$

$$x \leqslant^{l}_{L_2} y \Leftrightarrow L_2\,x^{-1} \subseteq L_2\,y^{-1}$$

$L_2 = \{ab, ac\}$ we have $b \leqslant^{l}_{L_2} c$ since $L_2\,b^{-1} = L_2\,c^{-1} = \{a\}$

Example of quasiorder

Nerode left quasiorder relative to L_2 [Varricchio,deLuca'94]

$$x \leqslant^{l}_{L_2} y \Leftrightarrow L_2\, x^{-1} \subseteq L_2\, y^{-1}$$

$L_2 = \{ab, ac\}$ we have $b \leqslant^{l}_{L_2} c$ since $L_2\, b^{-1} = L_2\, c^{-1} = \{a\}$

$\leqslant^{l}_{L_2}$ is left L_2 consistent

Example of quasiorder

Nerode left quasiorder relative to L_2 [Varricchio,deLuca'94]

$$x \leqslant^{l}_{L_2} y \Leftrightarrow L_2\, x^{-1} \subseteq L_2\, y^{-1}$$

$L_2 = \{ab, ac\}$ we have $b \leqslant^{l}_{L_2} c$ since $L_2\, b^{-1} = L_2\, c^{-1} = \{a\}$

$\leqslant^{l}_{L_2}$ is left L_2 consistent

$\leqslant^{l}_{L_2}$ is a well-quasi order

$\leqslant^{l}_{L_2}$ is decidable

} if L_2 is regular

Example of quasiorder

Nerode left quasiorder relative to L_2 [Varricchio,deLuca'94]

$$x \leqslant^{l}_{L_2} y \Leftrightarrow L_2\, x^{-1} \subseteq L_2\, y^{-1}$$

$L_2 = \{ab, ac\}$ we have $b \leqslant^{l}_{L_2} c$ since $L_2\, b^{-1} = L_2\, c^{-1} = \{a\}$

$\leqslant^{l}_{L_2}$ is left L_2 consistent

$\leqslant^{l}_{L_2}$ is a well-quasi order

$\leqslant^{l}_{L_2}$ is decidable

} if L_2 is regular

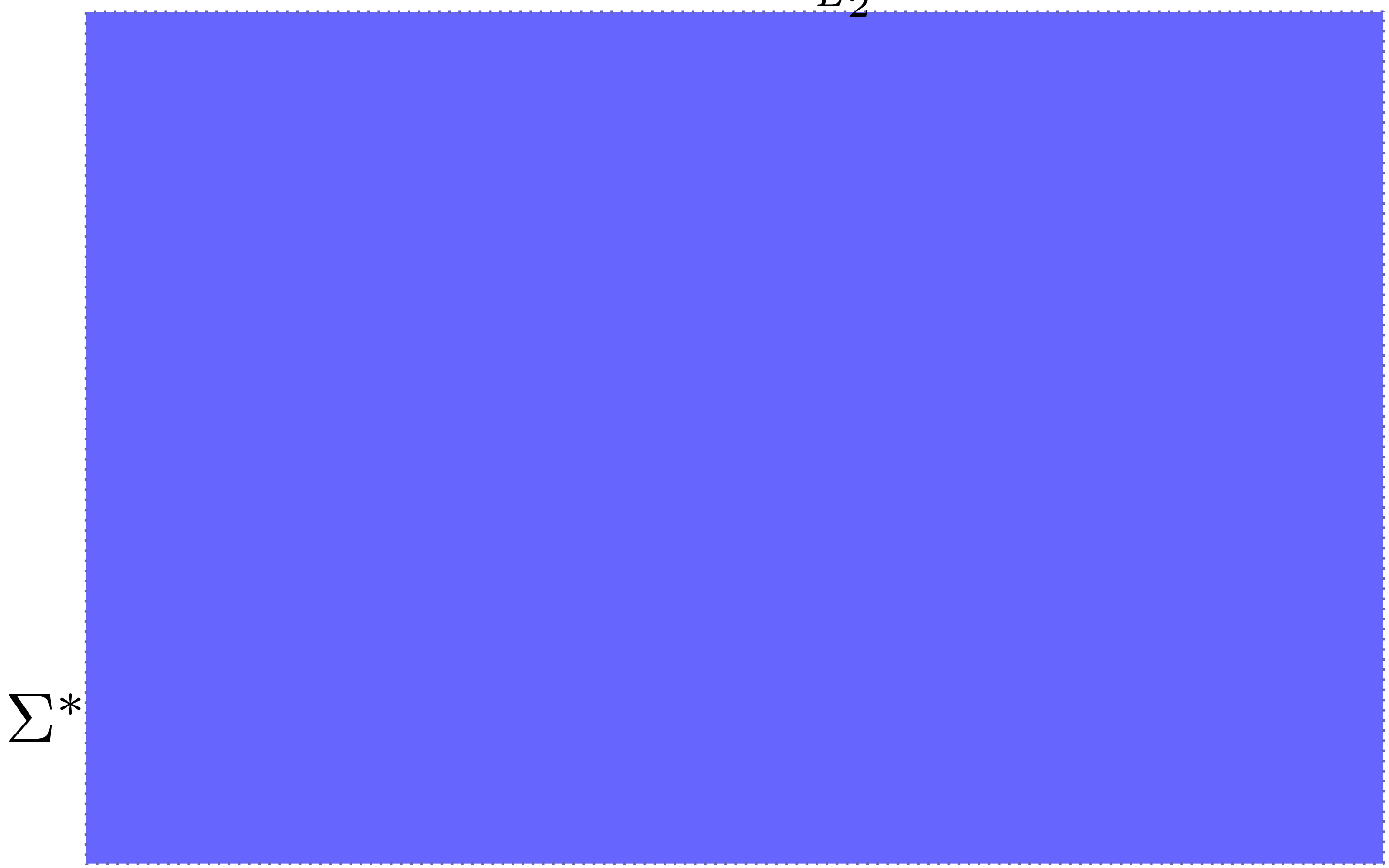
$\leqslant^{l}_{L_2}$ induce the coarsest qo based abstraction $\rho_{\leqslant^{l}_{L_2}}$

Deciding $L_1 \subseteq L_2$

$$L_1 \subseteq L_2 \quad \text{iff} \quad \rho_{\leqslant^l_{L_2}}(L_1) \subseteq L_2$$

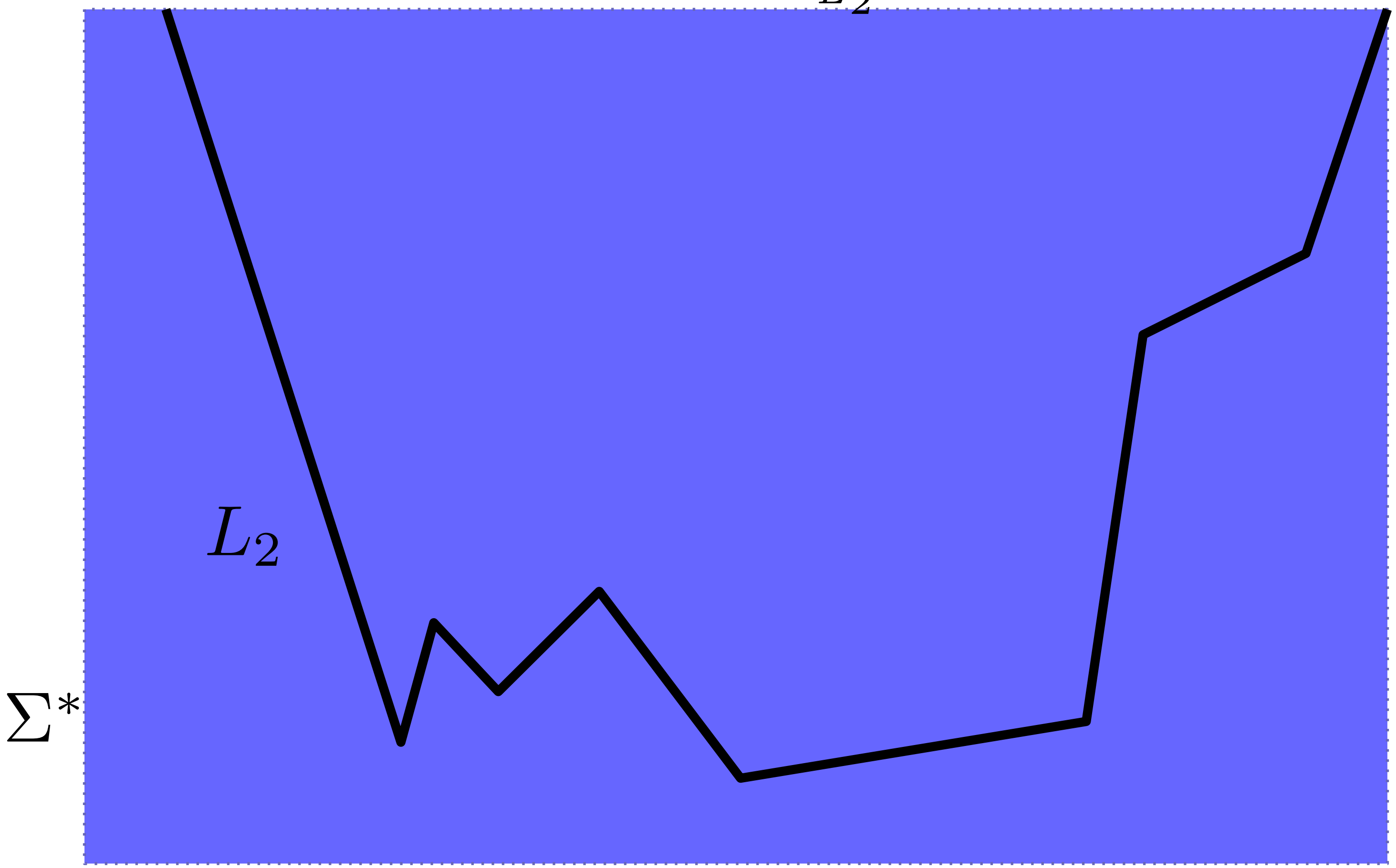
Deciding $L_1 \subseteq L_2$

$$L_1 \subseteq L_2 \quad \text{iff} \quad \rho_{\leqslant^l_{L_2}}(L_1) \subseteq L_2$$



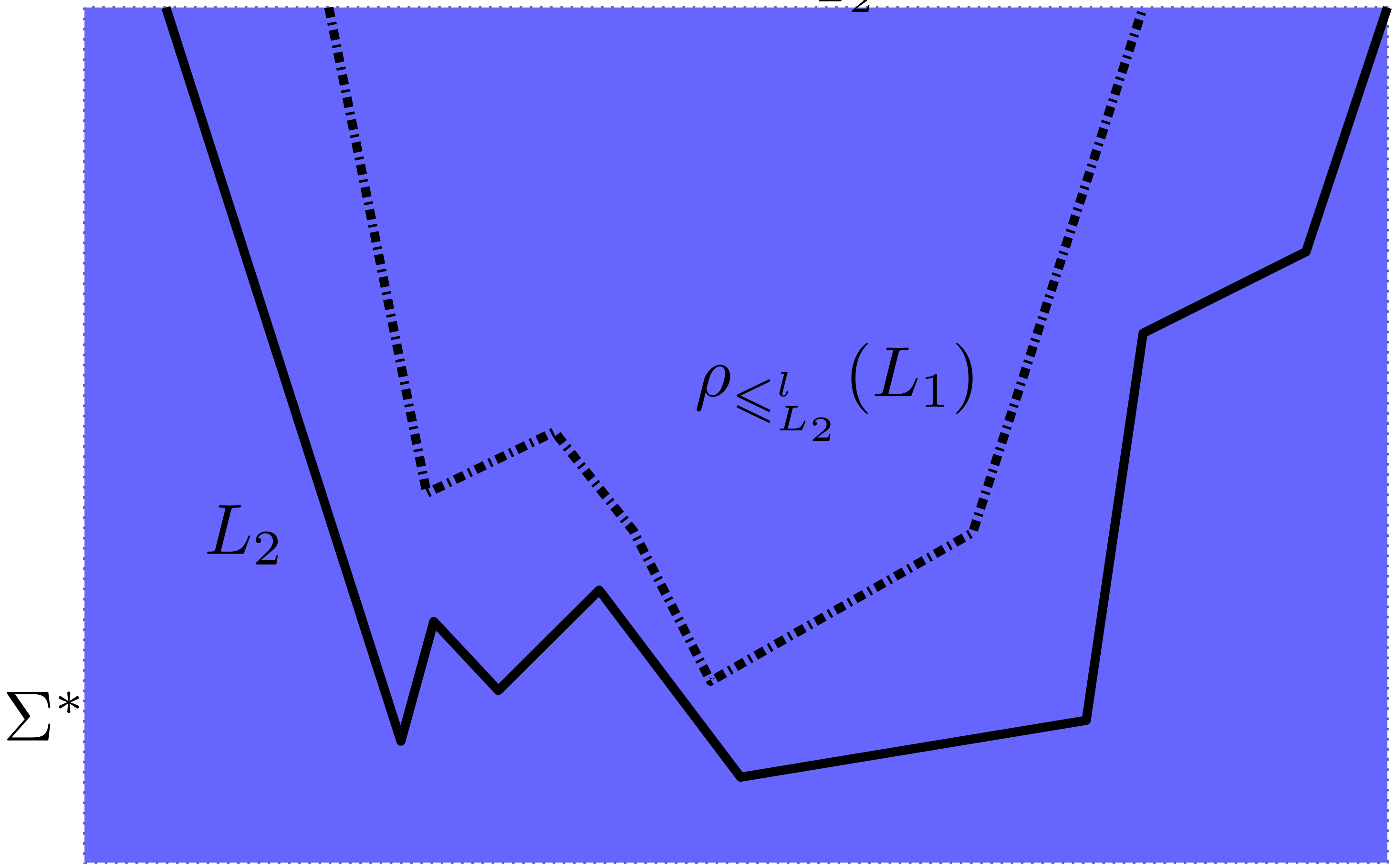
Deciding $L_1 \subseteq L_2$

$L_1 \subseteq L_2$ iff $\rho_{\leqslant^l_{L_2}}(L_1) \subseteq L_2$



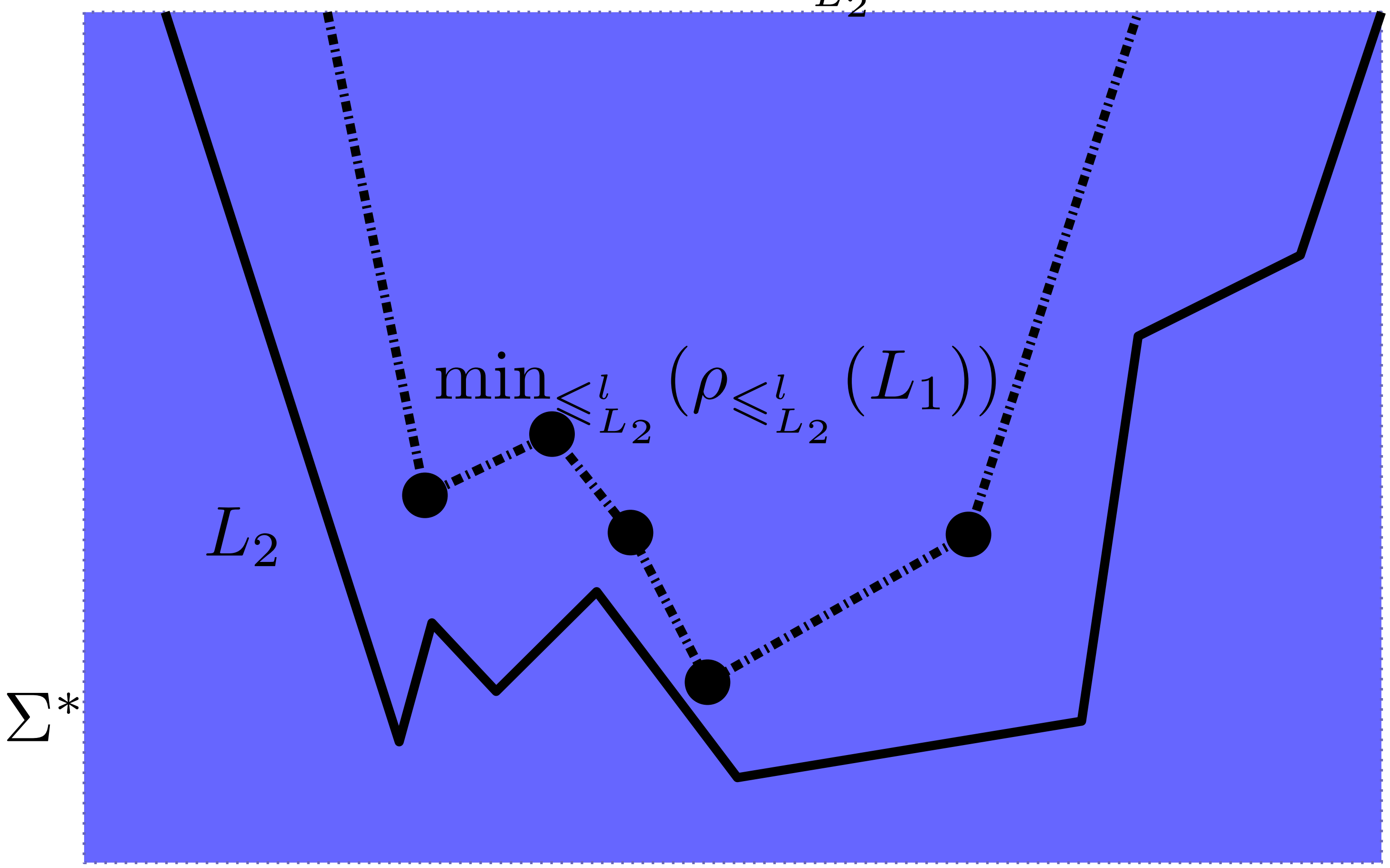
Deciding $L_1 \subseteq L_2$

$L_1 \subseteq L_2$ iff $\rho_{\leqslant^l_{L_2}}(L_1) \subseteq L_2$



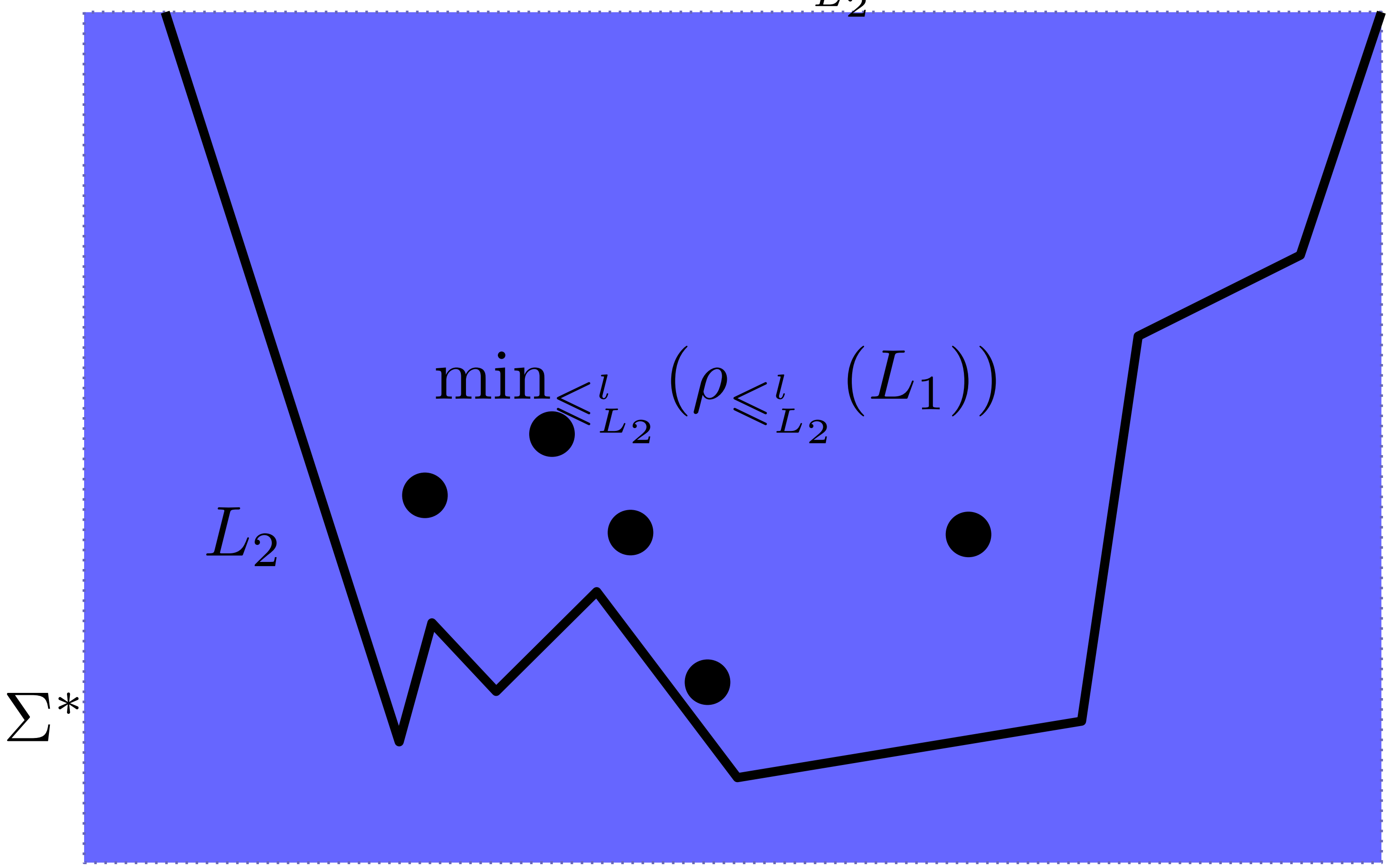
Deciding $L_1 \subseteq L_2$

$L_1 \subseteq L_2$ iff $\rho_{\leqslant^l_{L_2}}(L_1) \subseteq L_2$



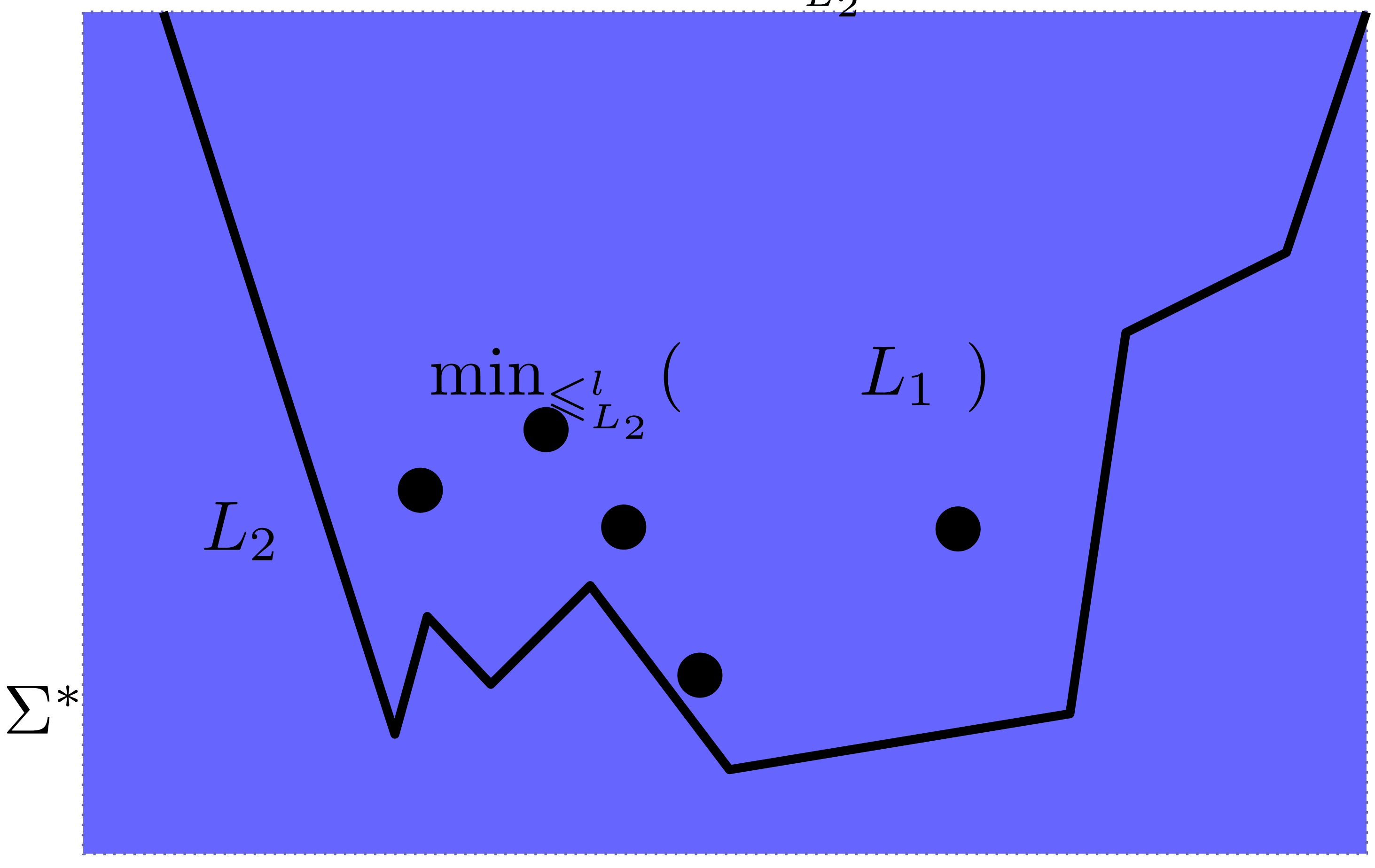
Deciding $L_1 \subseteq L_2$

$$L_1 \subseteq L_2 \quad \text{iff} \quad \rho_{\leqslant^l_{L_2}}(L_1) \subseteq L_2$$



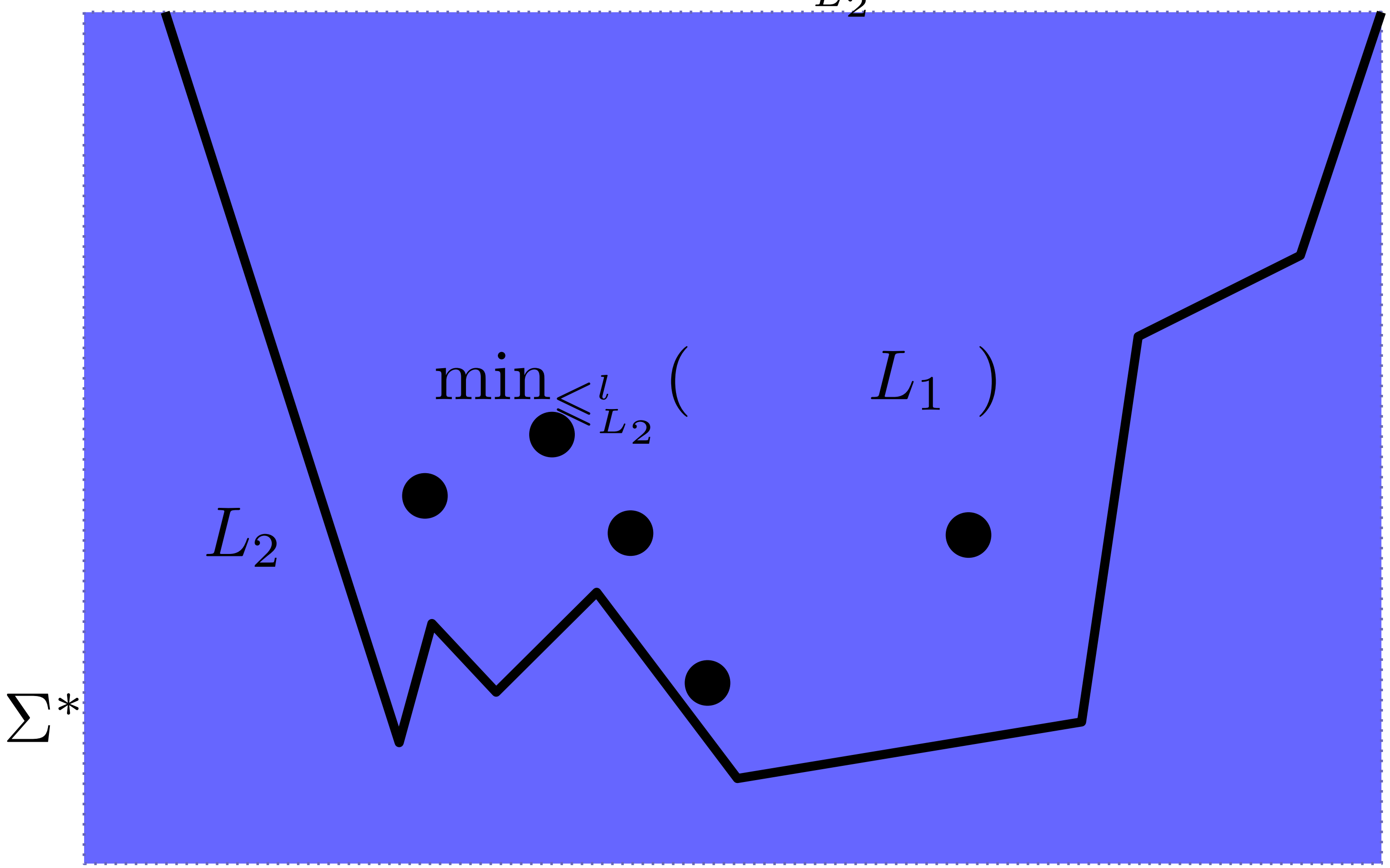
Deciding $L_1 \subseteq L_2$

$L_1 \subseteq L_2$ iff $\rho_{\leqslant^l_{L_2}}(L_1) \subseteq L_2$



Deciding $L_1 \subseteq L_2$

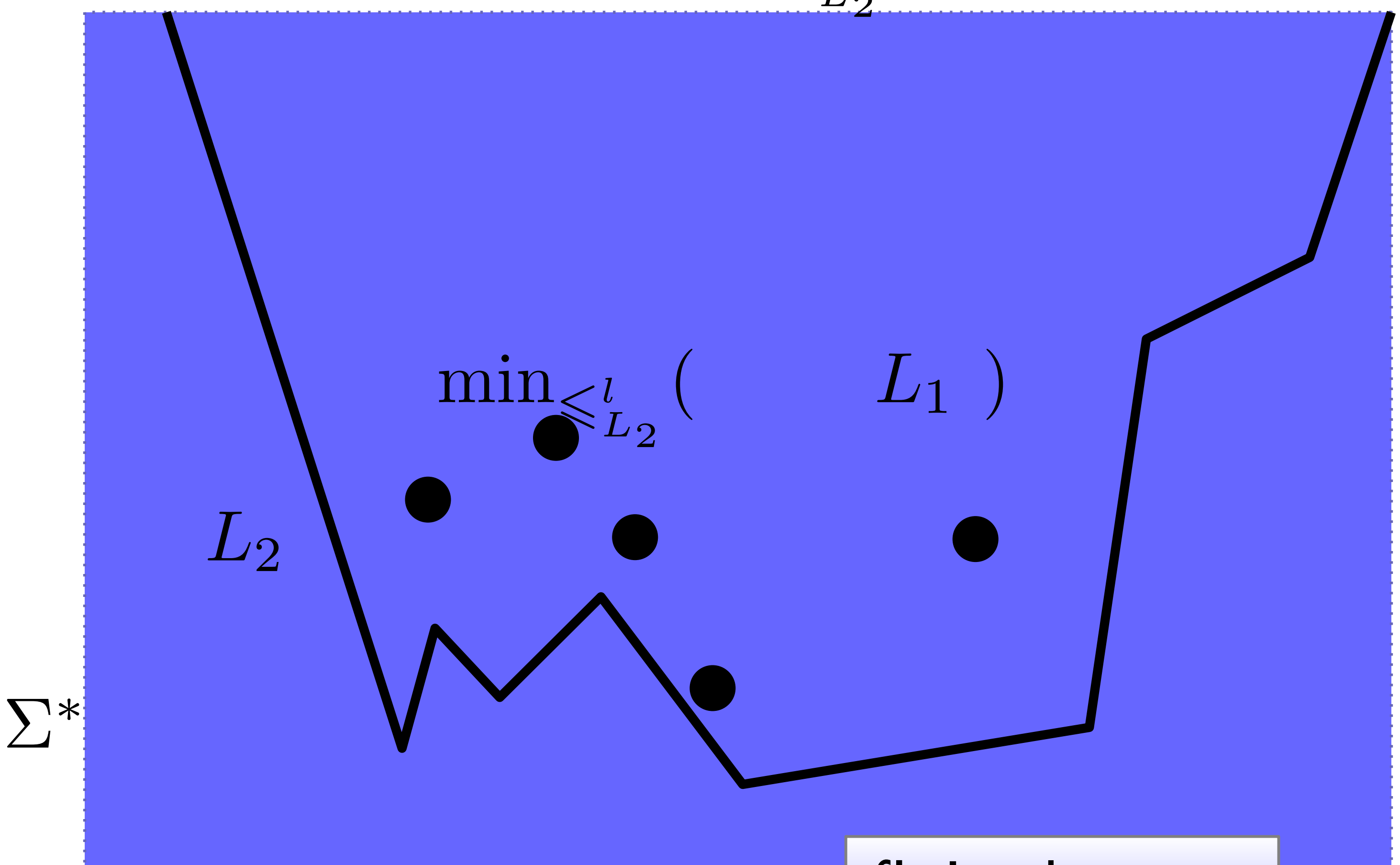
$$L_1 \subseteq L_2 \quad \text{iff} \quad \rho_{\leqslant^l_{L_2}}(L_1) \subseteq L_2$$



$$L_1 \subseteq L_2 \text{ iff } \forall w \in \min_{\leqslant^l_{L_2}}(L_1)\colon w \in L_2$$

Deciding $L_1 \subseteq L_2$

$L_1 \subseteq L_2$ iff $\rho_{\leqslant^l_{L_2}}(L_1) \subseteq L_2$



finite by wqo

$L_1 \subseteq L_2$ iff $\forall w \in \min_{\leqslant^l_{L_2}}(L_1)\colon w \in L_2$

Computing $\min_{\leqslant^l_{L_2}}(L_1)$

$$\min_{\leqslant^l_{L_2}}(L_1)$$

$$=$$

$$\min_{\leqslant^l_{L_2}}\left(\text{lfp}\,\lambda X.\,(b \cup Fn(X))\right)$$

$$=$$

$$\text{lfp}\,\lambda X.\,\min_{\leqslant^l_{L_2}}\left(b \cup Fn(X)\right)$$

Computing $\min_{\leqslant^{l}_{L_2}}(L_1)$

$$\min_{\leqslant^{l}_{L_2}}(L_1)$$

$$=$$

$$\min_{\leqslant^{l}_{L_2}}\big(\mathrm{lfp}\,\lambda X.\,(b \cup Fn(X))\big)$$

$$=$$

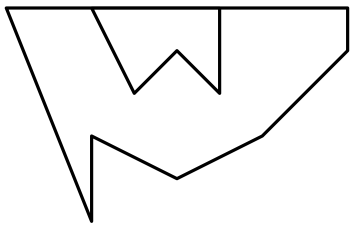
$$\mathrm{lfp}\,\lambda X.\,\min_{\leqslant^{l}_{L_2}}\big(b \cup Fn(X)\big)$$

Computing $\min_{\leqslant^l_{L_2}}(L_1)$

$$\min_{\leqslant^l_{L_2}}(L_1)$$

$$=$$

$$\min_{\leqslant^l_{L_2}}\big(\mathrm{lfp}\,\lambda X.\,(b \cup Fn(X))\big)$$



$$=$$

$$\mathrm{lfp}\,\lambda X.\,\min_{\leqslant^l_{L_2}}\big(b \cup Fn(X)\big)$$

Computing $\min_{\leqslant^l_{L_2}}(L_1)$

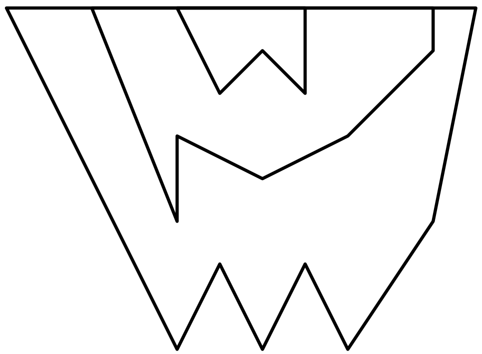
$$\min_{\leqslant^l_{L_2}}(L_1)$$

$$=$$

$$\min_{\leqslant^l_{L_2}}\big(\mathrm{lfp}\,\lambda X.\,(b \cup Fn(X))\big)$$

$$=$$

$$\mathrm{lfp}\,\lambda X.\,\min_{\leqslant^l_{L_2}}\big(b \cup Fn(X)\big)$$



Computing $\min_{\leqslant^l_{L_2}}(L_1)$

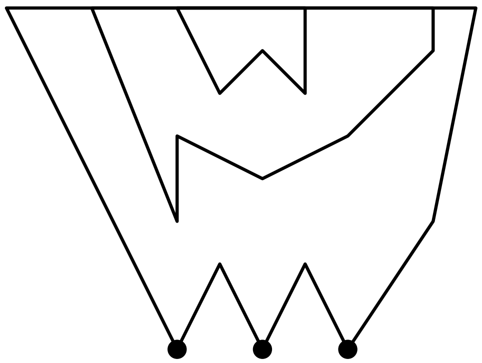
$$\min_{\leqslant^l_{L_2}}(L_1)$$

$$=$$

$$\min_{\leqslant^l_{L_2}}\big(\mathrm{lfp}\,\lambda X.\,(b \cup Fn(X))\big)$$

$$=$$

$$\mathrm{lfp}\,\lambda X.\,\min_{\leqslant^l_{L_2}}\big(b \cup Fn(X)\big)$$

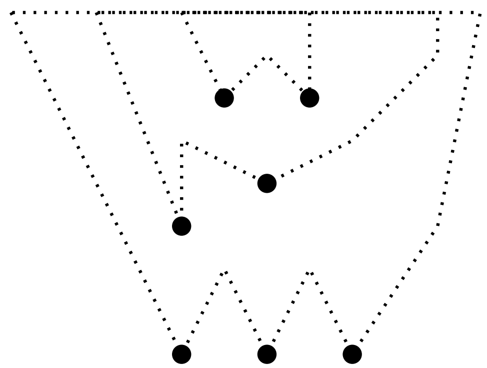


Computing $\min_{\leqslant^l_{L_2}}(L_1)$

$$\min_{\leqslant^l_{L_2}}(L_1)$$

$$=$$

$$\min_{\leqslant^l_{L_2}}\Big(\mathrm{lfp}\,\lambda X.\,(b \cup Fn(X))\Big)$$



$$=$$

$$\mathrm{lfp}\,\lambda X.\,\min_{\leqslant^l_{L_2}}\Big(b \cup Fn(X)\Big)$$

Word-based antichain algorithm for $\mathcal{L}(\mathcal{A}) \subseteq L_2$

```
Data: FA 𝒜 = ⟨Q, δ, q⁰, F⟩
Data: L₂ regular
⟨Y_q⟩_{q∈Q} := ∅⃗;
repeat
    ⟨X_q⟩_{q∈Q} := ⟨Y_q⟩_{q∈Q};
    ⟨Y_q⟩_{q∈Q} := min_{⩽^l_{L₂}} (b ∪ Fn(⟨Y_q⟩_{q∈Q}));
until ρ_{⩽^l_{L₂}}(⟨Y_q⟩_{q∈Q}) ⊆ ρ_{⩽^l_{L₂}}(⟨X_q⟩_{q∈Q});
forall u ∈ Y_{q⁰} do
    if u ∉ L₂ then return false;
return true;
```

What else?

Nerode left quasiorder relative to L_2

[Varricchio,deLuca'94]

$$x \leqslant^{l}_{L_2} y \Leftrightarrow L_2\, x^{-1} \subseteq L_2\, y^{-1}$$

What else?

Nerode left quasiorder relative to L_2

[Varricchio,deLuca'94]

$$x \leqslant^{l}_{L_2} y \Leftrightarrow L_2\, x^{-1} \subseteq L_2\, y^{-1}$$

State based quasiorder

Let $\mathcal{A}_2 = \langle Q_2, \delta_2, q_2^0, F_2 \rangle$ be an automaton for L_2

$$x \leqslant^{l}_{\mathcal{A}_2} y \Leftrightarrow \mathrm{pre}_x(F_2) \subseteq \mathrm{pre}_y(F_2)$$

What else?

Nerode left quasiorder relative to L_2

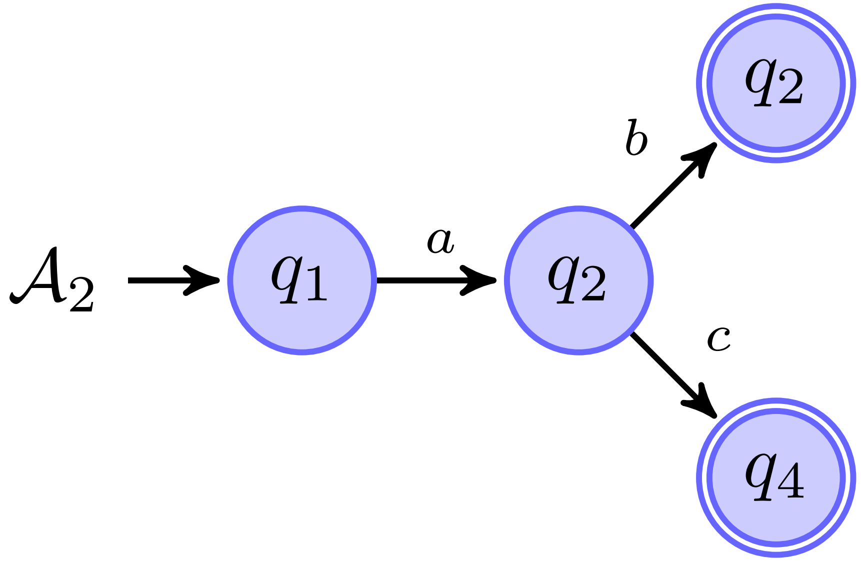
[Varricchio,deLuca'94]

$$x \leqslant^l_{L_2} y \Leftrightarrow L_2\, x^{-1} \subseteq L_2\, y^{-1}$$

State based quasiorder

Let $\mathcal{A}_2 = \langle Q_2, \delta_2, q_2^0, F_2 \rangle$ be an automaton for L_2

$$x \leqslant^l_{\mathcal{A}_2} y \Leftrightarrow \mathrm{pre}_x(F_2) \subseteq \mathrm{pre}_y(F_2)$$



What else?

Nerode left quasiorder relative to L_2

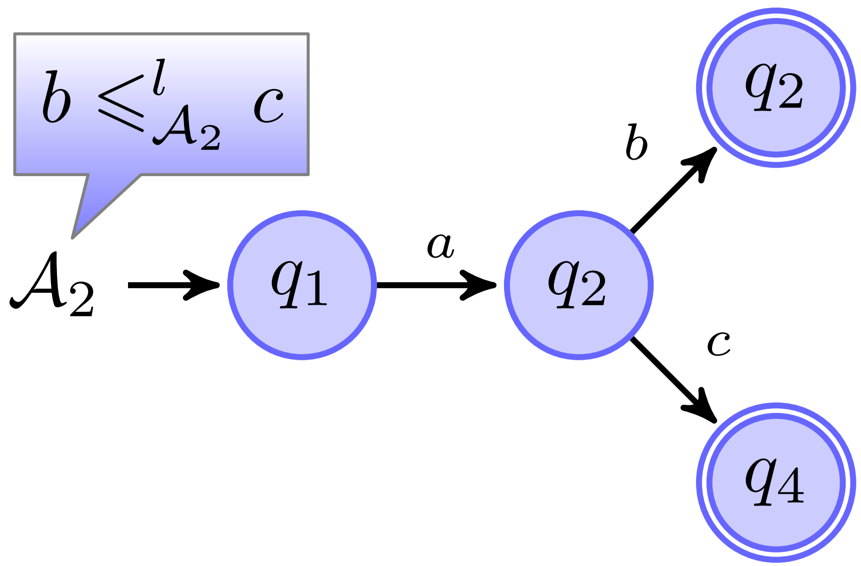
[Varricchio,deLuca'94]

$$x \leqslant^l_{L_2} y \Leftrightarrow L_2\, x^{-1} \subseteq L_2\, y^{-1}$$

State based quasiorder

Let $\mathcal{A}_2 = \langle Q_2, \delta_2, q_2^0, F_2 \rangle$ be an automaton for L_2

$$x \leqslant^l_{\mathcal{A}_2} y \Leftrightarrow \mathrm{pre}_x(F_2) \subseteq \mathrm{pre}_y(F_2)$$



What else?

Nerode left quasiorder relative to L_2

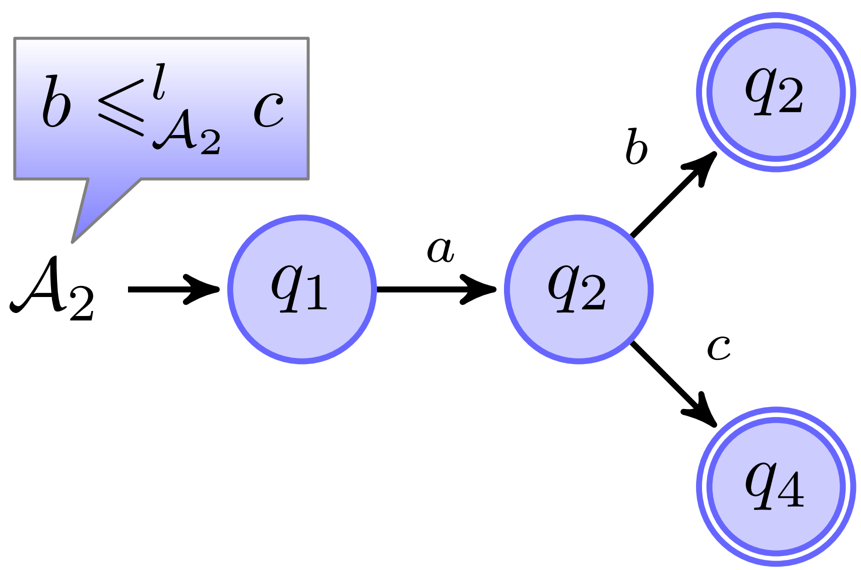
[Varricchio,deLuca'94]

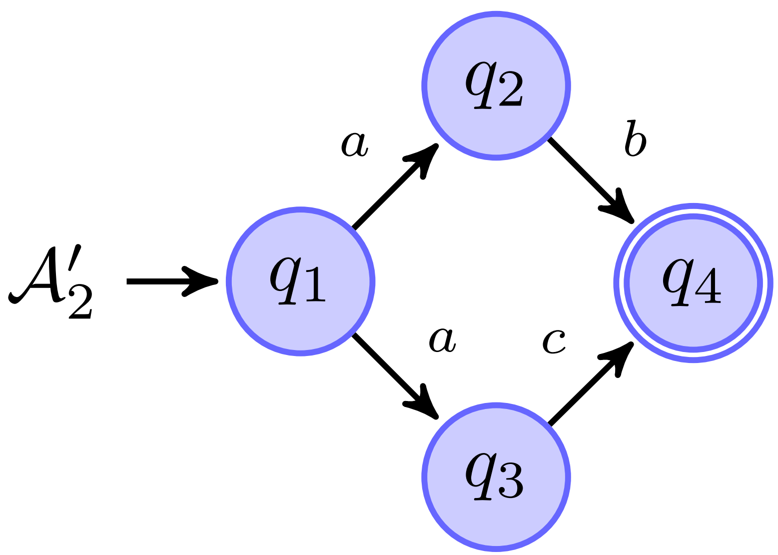
$$x \leqslant^l_{L_2} y \Leftrightarrow L_2\, x^{-1} \subseteq L_2\, y^{-1}$$

State based quasiorder

Let $\mathcal{A}_2 = \langle Q_2, \delta_2, q_2^0, F_2 \rangle$ be an automaton for L_2

$$x \leqslant^l_{\mathcal{A}_2} y \Leftrightarrow \mathrm{pre}_x(F_2) \subseteq \mathrm{pre}_y(F_2)$$





What else?

Nerode left quasiorder relative to L_2

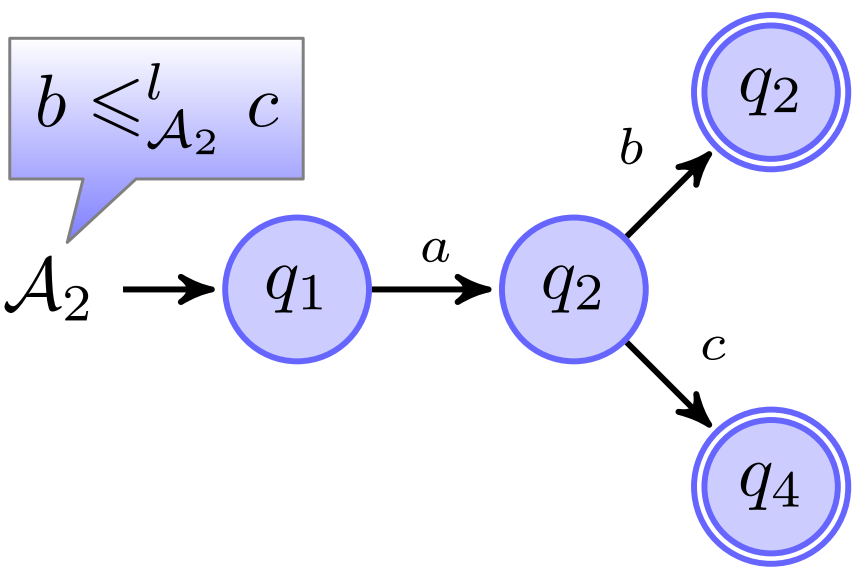
[Varricchio,deLuca'94]

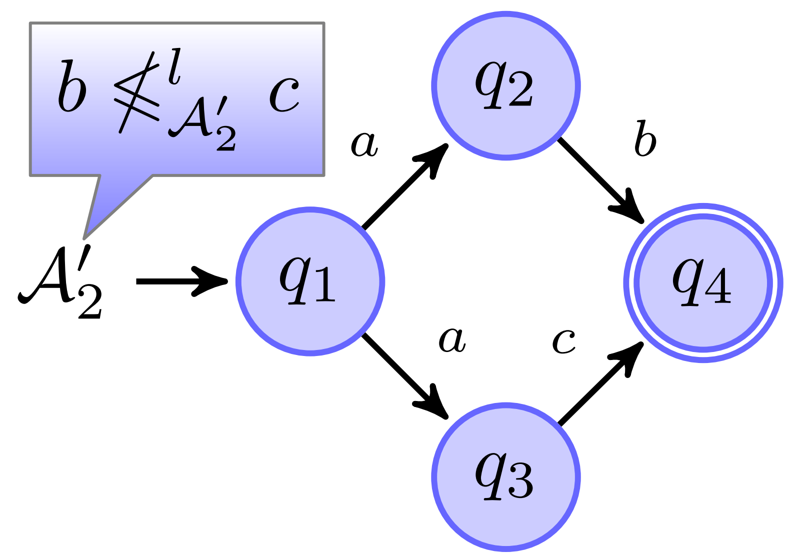
$$x \leqslant^{l}_{L_2} y \Leftrightarrow L_2\, x^{-1} \subseteq L_2\, y^{-1}$$

State based quasiorder

Let $\mathcal{A}_2 = \langle Q_2, \delta_2, q_2^0, F_2 \rangle$ be an automaton for L_2

$$x \leqslant^{l}_{\mathcal{A}_2} y \Leftrightarrow \mathrm{pre}_x(F_2) \subseteq \mathrm{pre}_y(F_2)$$





What else?

Nerode left quasiorder relative to L_2

[Varricchio,deLuca'94]

$$x \leqslant^l_{L_2} y \Leftrightarrow L_2\, x^{-1} \subseteq L_2\, y^{-1}$$

State based quasiorder

Let $\mathcal{A}_2 = \langle Q_2, \delta_2, q_2^0, F_2 \rangle$ be an automaton for L_2

$$x \leqslant^l_{\mathcal{A}_2} y \Leftrightarrow \mathrm{pre}_x(F_2) \subseteq \mathrm{pre}_y(F_2)$$

1. $\leqslant^l_{\mathcal{A}_2} \cap (L_2 \times \overline{L_2}) = \varnothing$
2. $x \leqslant^l_{\mathcal{A}_2} y$ implies $a\,x \leqslant^l_{\mathcal{A}_2} a\,y$ for all a
3. $\leqslant^l_{\mathcal{A}_2}$ is a well-quasiorder
4. $\leqslant^l_{\mathcal{A}_2}$ is decidable

What else?

Nerode left quasiorder relative to L_2 [Varricchio,deLuca'94]

$$x \leqslant^l_{L_2} y \Leftrightarrow L_2\, x^{-1} \subseteq L_2\, y^{-1}$$

State based quasiorder

Let $\mathcal{A}_2 = \langle Q_2, \delta_2, q_2^0, F_2 \rangle$ be an automaton for L_2

$$x \leqslant^l_{\mathcal{A}_2} y \Leftrightarrow \mathrm{pre}_x(F_2) \subseteq \mathrm{pre}_y(F_2)$$

1. $\leqslant^l_{\mathcal{A}_2} \cap (L_2 \times \overline{L_2}) = \varnothing$
2. $x \leqslant^l_{\mathcal{A}_2} y$ implies $a\,x \leqslant^l_{\mathcal{A}_2} a\,y$ fo
3. $\leqslant^l_{\mathcal{A}_2}$ is a well-quasiorder
4. $\leqslant^l_{\mathcal{A}_2}$ is decidable

$\leqslant^l_{\mathcal{A}_2}$ is left L_2 consistent decidable wqo

Word-based antichain algorithm for $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$

```
Data: FA A_1 = ⟨Q_1, δ_1, q_1^0, F_1⟩
Data: FA A_2 = ⟨Q_2, δ_2, q_2^0, F_2⟩
⟨Y_q⟩_{q∈Q_1} := ∅⃗;
repeat
    ⟨X_q⟩_{q∈Q_1} := ⟨Y_q⟩_{q∈Q_1};
    ⟨Y_q⟩_{q∈Q_1} := min_{⩽^l_{A_2}} (b ∪ Fn(⟨Y_q⟩_{q∈Q_1}));
until ρ_{⩽^l_{A_2}}(⟨Y_q⟩_{q∈Q_1}) ⊆ ρ_{⩽^l_{A_2}}(⟨X_q⟩_{q∈Q_1});
forall u ∈ Y_{q_1^0} do
    if u ∉ L(A_2) then return false;
return true;
```

Data: FA $\mathcal{A}_1 = \langle Q_1, \delta_1, q_1^0, F_1 \rangle$
Data: FA $\mathcal{A}_2 = \langle Q_2, \delta_2, q_2^0, F_2 \rangle$
$\langle Y_q \rangle_{q \in Q_1} := \vec{\varnothing}$;
repeat
    $\langle X_q \rangle_{q \in Q_1} := \langle Y_q \rangle_{q \in Q_1}$;
    $\langle Y_q \rangle_{q \in Q_1} := \min_{\leqslant^l_{\mathcal{A}_2}} \big(b \cup Fn(\langle Y_q \rangle_{q \in Q_1}) \big)$;
until $\rho_{\leqslant^l_{\mathcal{A}_2}}(\langle Y_q \rangle_{q \in Q_1}) \subseteq \rho_{\leqslant^l_{\mathcal{A}_2}}(\langle X_q \rangle_{q \in Q_1})$;
forall $u \in Y_{q_1^0}$ **do**
    if $u \notin L(\mathcal{A}_2)$ **then return** *false*;
return *true*;

Word-based antichain algorithm for $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$

Data: FA $\mathcal{A}_1 = \langle Q_1, \delta_1, q_1^0, F_1 \rangle$
Data: FA $\mathcal{A}_2 = \langle Q_2, \delta_2, q_2^0, F_2 \rangle$

$\langle Y_q \rangle_{q \in Q_1} := \vec{\varnothing}$;

State based quasiorder

Let $\mathcal{A}_2 = \langle Q_2, \delta_2, q_2^0, F_2 \rangle$ be an automaton for L_2

$$x \leqslant^{l}_{\mathcal{A}_2} y \Leftrightarrow \mathrm{pre}_x(F_2) \subseteq \mathrm{pre}_y(F_2)$$

if $u \notin L(\mathcal{A}_2)$ **then return** *false*;

return *true*;

Ditching words altogether

State-based antichain algorithm for $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$

Data: FA $\mathcal{A}_1 = \langle Q_1, \delta_1, q_1^0, F_1 \rangle$
Data: FA $\mathcal{A}_2 = \langle Q_2, \delta_2, q_2^0, F_2 \rangle$

$\langle Y_q \rangle_{q \in Q_1} := \vec{\varnothing}$;
repeat
    $\langle X_q \rangle_{q \in Q_1} := \langle Y_q \rangle_{q \in Q_1}$;
    $\langle Y_q \rangle_{q \in Q_1} := [\ldots]$;
until $Y_q \subseteq^{\forall\exists} X_q$, *for all* $q \in Q_1$;
forall $s \in Y_{q_1^0}$ **do**
    if $q_2^0 \notin s$ **then return** *false*;
return *true*;

$$[\ldots] = \min_{\subseteq^{\forall\exists}} \{ \mathrm{pre}_a^{\mathcal{A}_2}(s) \mid \exists a \in \Sigma,\ q' \in \delta_1(q, a), s \in X_{q'} \} \cup F_2$$

- Same as antichain algo [1]
- Derived from general solution instantiated to $\leqslant^{l}_{\mathcal{A}_2}$
- Variants of the antichain algorithm as instantiations

[1] De Wulf, Doyen, Henzinger, and Raskin. *"Antichains: A New Algorithm for Checking Universality of Finite Automata."* In CAV'06. (cited 171 times)

- Same as antichain algo [1]
- Derived from general solution instantiated to $\leqslant^{l}_{\mathcal{A}_2}$
- Variants of the antichain algorithm as instantiations
- Simulation enhanced antichain algo with simulation enhanced qo [2*]

[1] De Wulf, Doyen, Henzinger, and Raskin. *"Antichains: A New Algorithm for Checking Universality of Finite Automata."* In CAV'06. (cited 171 times)

[2] Abdulla, Chen, Holík, Mayr, and Vojnar. *"When Simulation Meets Antichains."* In TACAS'10.
Bonchi, and Pous. *"Checking NFA Equivalence with Bisimulations up to Congruence."* In POPL'13.

$Reg \subseteq Reg$

1. $\leqslant^{l}_{L_2} \cap (L_2 \times \overline{L_2}) = \varnothing$
2. $x \leqslant^{l}_{L_2} y$ implies $a\,x \leqslant^{l}_{L_2} a\,y$ for all a
3. $\leqslant^{l}_{L_2}$ is a well-quasiorder
4. $\leqslant^{l}_{L_2}$ is decidable

(3 and 4:) if L_2 is regular

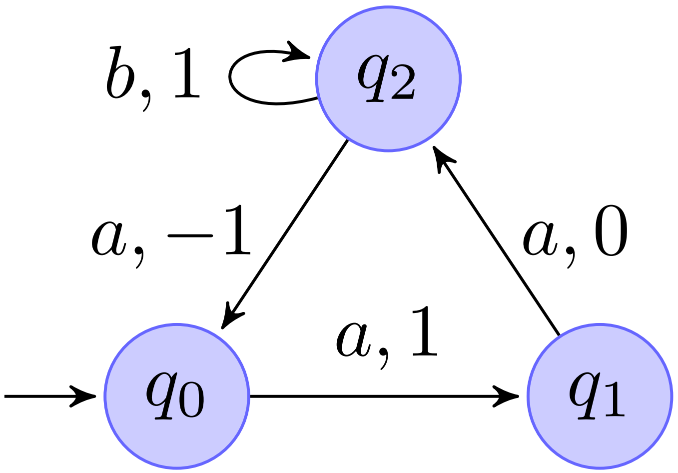
$$L_1 \subseteq L_2 \text{ iff } \operatorname{lfp} \lambda X.\, \min_{\leqslant^{l}_{L_2}} \big(b \cup Fn(X)\big) \subseteq L_2$$

if L_2 is regular then $\leqslant^{l}_{L_2}$ is a wqo

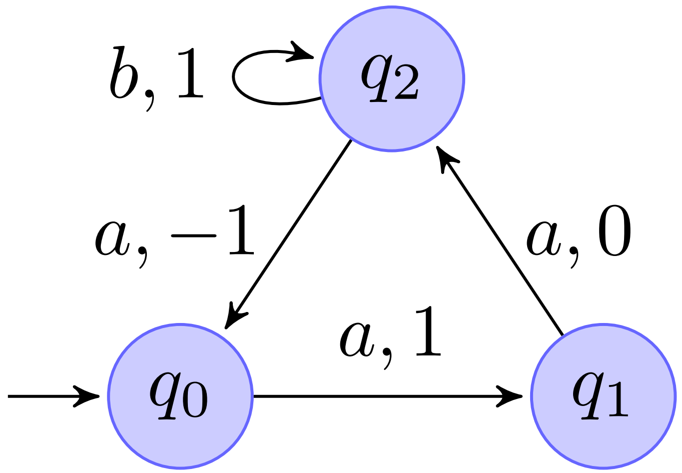
but the opposite might not hold

$Reg \subseteq BeyondRegular$

L_2 is a one-counter net trace set



L_2 is a one-counter net trace set

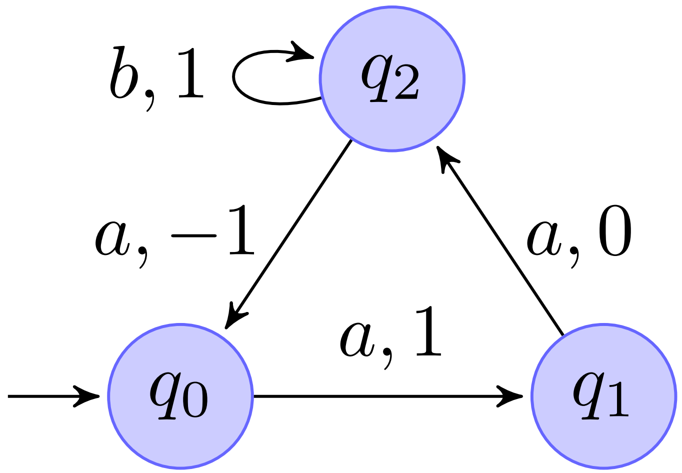


$L_2 = \mathcal{L}(\mathcal{O}_2)$

Traces from configuration

$q_0\,0$

L_2 is a one-counter net trace set

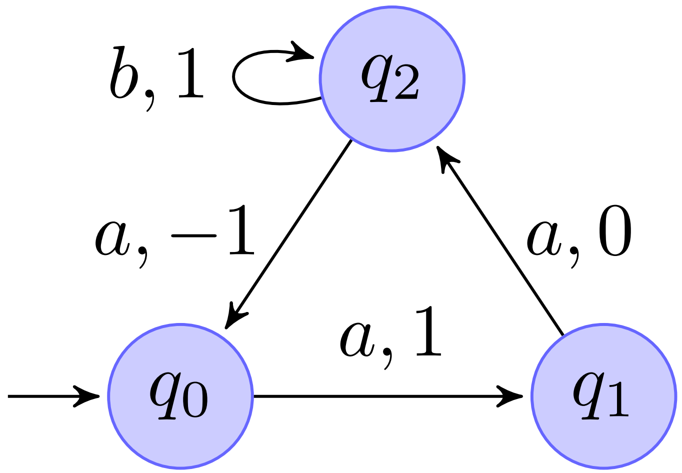


$L_2 = \mathcal{L}(\mathcal{O}_2)$

Traces from configuration $q_0\, 0$

The right Nerode qo for L_2 ($\leqslant^r_{L_2}$) is a wqo but it's undecidable

L_2 is a one-counter net trace set



$L_2 = \mathcal{L}(\mathcal{O}_2)$

Traces from configuration $q_0\,0$

The right Nerode qo for L_2 ($\leqslant^r_{L_2}$) is a wqo but it's undecidable

$u \leqslant^r_{\mathcal{O}_2} v$ iff $m_u \leq m_v$ where $m_u, m_v \in (\mathbb{N} \cup \{\perp\})^3$

$u \leqslant^r_{\mathcal{O}_2} v$ is a right L_2 consistent decidable wqo

Decision procedure for $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{O}_2)$ using $\leqslant^r_{\mathcal{O}_2}$

Word-based antichain algorithm for $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{O}_2)$

Data: FA $\mathcal{A}_1 = \langle Q, \delta, q^0, F \rangle$
Data: OCN $\mathcal{O}_2$

$\langle Y_q \rangle_{q \in Q} := \vec{\varnothing}$;
repeat
    $\langle X_q \rangle_{q \in Q} := \langle Y_q \rangle_{q \in Q}$;
    $\langle Y_q \rangle_{q \in Q} := \min_{\leqslant^r_{\mathcal{O}_2}} \big(b \cup Fn(\langle Y_q \rangle_{q \in Q})\big)$;
until $\rho_{\leqslant^r_{\mathcal{O}_2}}(\langle Y_q \rangle_{q \in Q}) \subseteq \rho_{\leqslant^r_{\mathcal{O}_2}}(\langle X_q \rangle_{q \in Q})$;
forall $u \in Y_{q^0}$ **do**
    if $u \notin L_2$ **then return** *false*;
return *true*;

Dual approach

$$\mathrm{lfp}\,\lambda X.\, b \cup Fn(X) \subseteq L_2 \quad \text{iff} \quad b \subseteq \mathrm{gfp}\,\lambda X.\, L_2 \cap \widetilde{Fn}(X)$$

Dual approach

$$\mathrm{lfp}\,\lambda X.\, b \cup Fn(X) \subseteq L_2 \quad \text{iff} \quad b \subseteq \mathrm{gfp}\,\lambda X.\, L_2 \cap \widetilde{Fn}(X)$$

$\cup$ becomes $\cap$

$\lambda X.\, aX$ becomes $\lambda X.\, a^{-1}X$

$\rho\ Fn\ \rho = \rho\ Fn$ becomes $\rho\ \widetilde{Fn}\ \rho = \widetilde{Fn}\ \rho$

Dual approach

Does not terminate in general

$$\mathrm{lfp}\,\lambda X.\, b \cup Fn(X) \subseteq L_2 \quad \text{iff} \quad b \subseteq \mathrm{gfp}\,\lambda X.\, L_2 \cap \widetilde{Fn}(X)$$

$\cup$ becomes $\cap$

$\lambda X.\, aX$ becomes $\lambda X.\, a^{-1}X$

$\rho\ Fn\ \rho = \rho\ Fn$ becomes $\rho\ \widetilde{Fn}\ \rho = \widetilde{Fn}\ \rho$

Dual approach

Does not terminate in general

Does always terminate [1]

$$\operatorname{lfp} \lambda X.\, b \cup Fn(X) \subseteq L_2 \quad \text{iff} \quad b \subseteq \operatorname{gfp} \lambda X.\, L_2 \cap \widetilde{Fn}(X)$$

$\cup$ becomes $\cap$

$\lambda X.\, aX$ becomes $\lambda X.\, a^{-1}X$

$\rho\ Fn\ \rho = \rho\ Fn$ becomes $\rho\ \widetilde{Fn}\ \rho = \widetilde{Fn}\ \rho$

Dual approach

Does not terminate in general

Does always terminate [1]

$$\mathrm{lfp}\,\lambda X.\, b \cup Fn(X) \subseteq L_2 \quad \text{iff} \quad b \subseteq \mathrm{gfp}\,\lambda X.\, L_2 \cap \widetilde{Fn}(X)$$

$\cup$ becomes $\cap$

$\lambda X.\, aX$ becomes $\lambda X.\, a^{-1}X$

$\rho\ Fn\ \rho = \rho\ Fn$ becomes $\rho\ \widetilde{Fn}\ \rho = \widetilde{Fn}\ \rho$

$\mathrm{gfp}\,\lambda X.\, \rho(L_2 \cap \widetilde{Fn}(X))$ stepwise equal $\mathrm{gfp}\,\lambda X.\, L_2 \cap \widetilde{Fn}(X)$

Dual approach

Does not terminate in general

Does always terminate [1]

$$\mathrm{lfp}\,\lambda X.\, b \cup Fn(X) \subseteq L_2 \quad \text{iff} \quad b \subseteq \mathrm{gfp}\,\lambda X.\, L_2 \cap \widetilde{Fn}(X)$$

$$\cup \quad \text{becomes} \quad \cap$$

$$\lambda X.\, aX \quad \text{becomes} \quad \lambda X.\, a^{-1}X$$

$$\rho\ Fn\ \rho = \rho\ Fn \quad \text{becomes} \quad \rho\ \widetilde{Fn}\ \rho = \widetilde{Fn}\ \rho$$

$$\mathrm{gfp}\,\lambda X.\, \rho(L_2 \cap \widetilde{Fn}(X)) \quad \text{stepwise equal} \quad \mathrm{gfp}\,\lambda X.\, L_2 \cap \widetilde{Fn}(X)$$

new algos

Quasiorders for the case $\mathcal{L}(\mathcal{G}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$

$Fn_{\mathcal{G}_1}$ is more general: from aX, Xa, aXb, XX

Quasiorders for the case $\mathcal{L}(\mathcal{G}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$

$Fn_{\mathcal{G}_1}$ is more general: from aX, Xa, aXb, XX

$\rho(L_1) = \operatorname{lfp} \lambda X.\, \rho\big(b \cup Fn_{\mathcal{G}_1}(X)\big)$
if ρ is complete for $\lambda X.\, aX$ **and** $\lambda X.\, Xa$ for all a

Quasiorders for the case $\mathcal{L}(\mathcal{G}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$

$Fn_{\mathcal{G}_1}$ is more general: from aX, Xa, aXb, XX

$\rho(L_1) = \operatorname{lfp} \lambda X.\, \rho\big(b \cup Fn_{\mathcal{G}_1}(X)\big)$
if ρ is complete for $\lambda X.\, aX$ **and** $\lambda X.\, Xa$ for all a

1. $\leqslant \cap \, (L_2 \times \overline{L_2}) = \varnothing$
2. **2.** $x \leqslant y$ implies $a\,x \leqslant a\,y$ **and** $x\,a \leqslant y\,a$ for all a

Quasiorders for the case $\mathcal{L}(\mathcal{G}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$

$Fn_{\mathcal{G}_1}$ is more general: from aX, Xa, aXb, XX

$\rho(L_1) = \operatorname{lfp} \lambda X.\, \rho\big(b \cup Fn_{\mathcal{G}_1}(X)\big)$
if ρ is complete for $\lambda X.\, aX$ **and** $\lambda X.\, Xa$ for all a

1. $\leqslant \cap\, (L_2 \times \overline{L_2}) = \varnothing$
2. $x \leqslant y$ implies $a\, x \leqslant a\, y$ **and** $x\, a \leqslant y\, a$ for all a

Myhill quasiorder relative to L_2 [Varricchio, deLuca'94]

$$x \leqslant_{L_2} y \Leftrightarrow \{(u,v) \mid u\, x\, v \in L_2\} \subseteq \{(u,v) \mid u\, y\, v \in L_2\}$$

Quasiorders for the case $\mathcal{L}(\mathcal{G}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$

$Fn_{\mathcal{G}_1}$ is more general: from aX, Xa, aXb, XX

$\rho(L_1) = \operatorname{lfp} \lambda X.\, \rho\big(b \cup Fn_{\mathcal{G}_1}(X)\big)$
if ρ is complete for $\lambda X.\, aX$ **and** $\lambda X.\, Xa$ for all a

1. $\leqslant \cap \, (L_2 \times \overline{L_2}) = \varnothing$
2. $x \leqslant y$ implies $a\, x \leqslant a\, y$ **and** $x\, a \leqslant y\, a$ for all a

Myhill quasiorder relative to L_2 [Varricchio, deLuca'94]

$$x \leqslant_{L_2} y \Leftrightarrow \{(u,v) \mid u\, x\, v \in L_2\} \subseteq \{(u,v) \mid u\, y\, v \in L_2\}$$

State based quasi order [Holík, Meyer @ Netys'15]

$$x \leqslant_{\mathcal{A}_2} y \Leftrightarrow \{(q,q') \mid q \overset{x}{\rightsquigarrow} q'\} \subseteq \{(q,q') \mid q \overset{y}{\rightsquigarrow} q'\}$$

Conclusions

Complete Abstract Interpretations for

$Reg \subseteq Reg$ $\quad$ $Reg \subseteq OCN$ $\quad$ $CFL \subseteq Reg$

Conclusions

Complete Abstract Interpretations for

$$Reg \subseteq Reg \qquad Reg \subseteq OCN \qquad CFL \subseteq Reg$$

Systematic derivation of algorithms

- ▷ Nerode wqo, (macro)state-based wqo, ...
- ▷ lfp and gfp
- ▷ New perspective on antichains algorithms

Conclusions

Complete Abstract Interpretations for

$Reg \subseteq Reg$ $\quad$ $Reg \subseteq OCN$ $\quad$ $CFL \subseteq Reg$

Systematic derivation of algorithms

- ▷ Nerode wqo, state-based wqo, . . .
- ▷ lfp and gfp
- ▷ New perspective on antichains algorithms

new

Conclusions

Complete Abstract Interpretations for

$$Reg \subseteq Reg \qquad Reg \subseteq OCN \qquad CFL \subseteq Reg$$

Systematic derivation of algorithms

- ▷ Nerode wqo, (macro)state-based wqo, . . .
- ▷ lfp and gfp
- ▷ New perspective on antichains algorithms

Technical Details

- ▷ SAS 2019
- ▷ arxiv 1904.01388

Future Work

Future Work

ω-languages: ω-$CFL \subseteq \omega$-Reg

▷ reduction $\mathrm{gfp} \mapsto \mathrm{lfp}$ using ultimately periodic words

Future Work

ω-languages: ω-$CFL \subseteq \omega$-Reg

- reduction $\mathrm{gfp} \mapsto \mathrm{lfp}$ using ultimately periodic words

Regular tree languages

- On the MyHill-Nerode Theorem for Trees by Kozen

Future Work

ω-languages: $\omega\text{-}CFL \subseteq \omega\text{-}Reg$

$\triangleright$ reduction $\mathrm{gfp} \mapsto \mathrm{lfp}$ using ultimately periodic words

Regular tree languages

$\triangleright$ On the MyHill-Nerode Theorem for Trees by Kozen

Timed Languages (TA $\subseteq$ 1-clock TA)

$\triangleright$ decidability by Ouaknine, Worrell (LICS 2004); reduction to emptiness of weakly alternating timed automata by Lasota, Walukiewicz (ACM ToCL 2008)