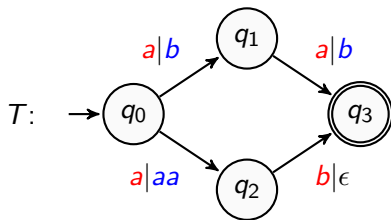


On Equivalence and Uniformisation Problems for Finite Transducers

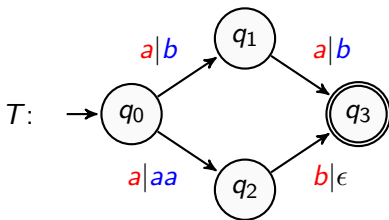
Emmanuel Filiot [ULB & FNRS],
Ismaël Jecker [ULB & FNRS],
Christof Löding [RWTH Aachen],
Sarah Winter [RWTH Aachen].

February 9, 2017

Transducer

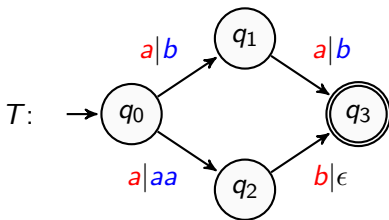


Transducer



A transducer T defines a relation $\mathcal{R}(T) \subseteq \Sigma^* \times \Sigma^*$.

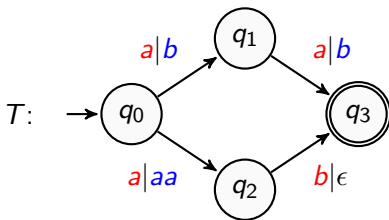
Transducer



$$\mathcal{R}(T) = \{(aa, bb), (ab, aa)\}.$$

A transducer T defines a relation $\mathcal{R}(T) \subseteq \Sigma^* \times \Sigma^*$.

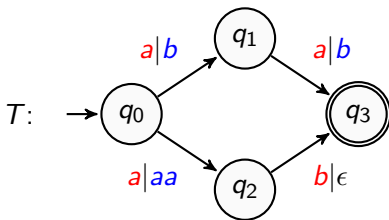
Transducer



$$\mathcal{R}(T) = \{(aa, bb), (ab, aa)\}.$$

A transducer T defines a relation $\mathcal{R}(T) \subseteq \Sigma^* \times \Sigma^*$.
The set of relations defined by transducers is RAT .

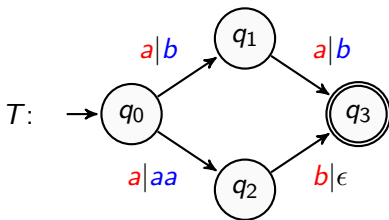
Transducer



$$\mathcal{R}(T) = \mathcal{F}(T) = \{(aa, bb), (ab, aa)\}.$$

A transducer T defines a relation $\mathcal{R}(T) \subseteq \Sigma^* \times \Sigma^*$.
The set of relations defined by transducers is RAT .

Transducer



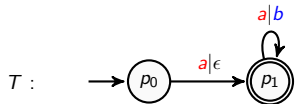
$$\mathcal{R}(T) = \mathcal{F}(T) = \{(aa, bb), (ab, aa)\}.$$

A transducer T defines a relation $\mathcal{R}(T) \subseteq \Sigma^* \times \Sigma^*$.

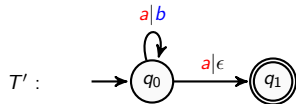
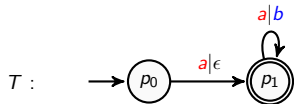
The set of relations defined by transducers is RAT .

The set of functions defined by transducers is $fRAT$.

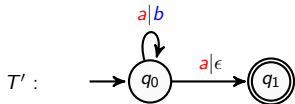
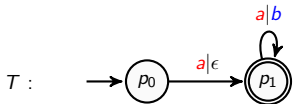
Inclusion and Equivalence Problems



Inclusion and Equivalence Problems



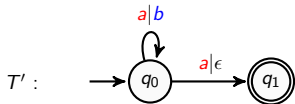
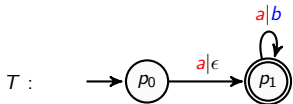
Inclusion and Equivalence Problems



Equivalence

$$\mathcal{R}(T) = \mathcal{R}(T')$$

Inclusion and Equivalence Problems

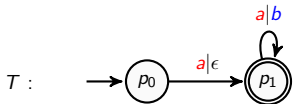


$$\mathcal{R}(T) = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$

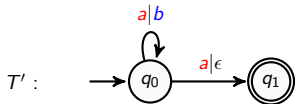
Equivalence

$$\mathcal{R}(T) = \mathcal{R}(T')$$

Inclusion and Equivalence Problems



$$\mathcal{R}(T) = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$

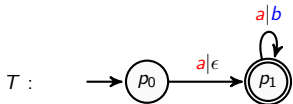


$$\mathcal{R}(T') = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$

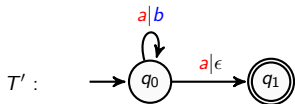
Equivalence

$$\mathcal{R}(T) = \mathcal{R}(T')$$

Inclusion and Equivalence Problems



$$\mathcal{R}(T) = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$



$$\mathcal{R}(T') = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$

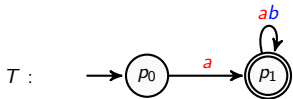
Equivalence

$$\mathcal{R}(T) = \mathcal{R}(T')$$

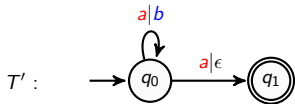
Undecidable¹

¹Timothy V. Griffiths, 1968.

Inclusion and Equivalence Problems



$$\mathcal{R}(T) = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$



$$\mathcal{R}(T') = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$

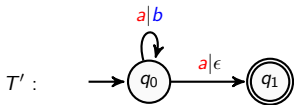
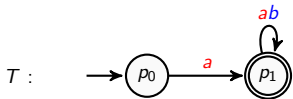
Equivalence

$$\mathcal{R}(T) = \mathcal{R}(T')$$

Undecidable¹

¹Timothy V. Griffiths, 1968.

Inclusion and Equivalence Problems



$$\mathcal{R}(T) = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$

$$\mathcal{L}(T) = \{a(ab)^n | n \in \mathbb{N}\}.$$

$$\mathcal{R}(T') = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$

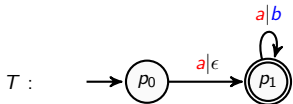
Equivalence

$$\mathcal{R}(T) = \mathcal{R}(T')$$

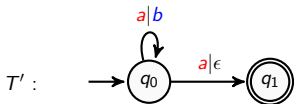
Undecidable¹

¹Timothy V. Griffiths, 1968.

Inclusion and Equivalence Problems



$$\mathcal{R}(T) = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$
$$\mathcal{L}(T) = \{a(ab)^n | n \in \mathbb{N}\}.$$



$$\mathcal{R}(T') = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$
$$\mathcal{L}(T') = \{(ab)^n a | n \in \mathbb{N}\}.$$

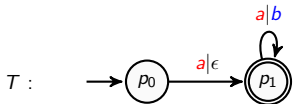
Equivalence

$$\mathcal{R}(T) = \mathcal{R}(T')$$

Undecidable¹

¹Timothy V. Griffiths, 1968.

Inclusion and Equivalence Problems

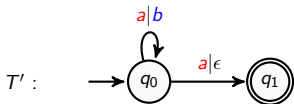


$$\mathcal{R}(T) = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$
$$\mathcal{L}(T) = \{a(ab)^n | n \in \mathbb{N}\}.$$

Equivalence

$$\mathcal{R}(T) = \mathcal{R}(T')$$

Undecidable¹



$$\mathcal{R}(T') = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$
$$\mathcal{L}(T') = \{(ab)^n a | n \in \mathbb{N}\}.$$

Origin Equivalence²

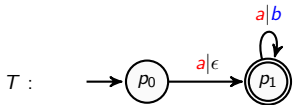
$$\mathcal{L}(T) = \mathcal{L}(T')$$

Decidable

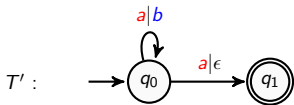
¹Timothy V. Griffiths, 1968.

²Mikołaj Bojańczyk, 2014.

Inclusion and Equivalence Problems



$$\mathcal{R}(T) = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$
$$\mathcal{L}(T) = \{a(ab)^n | n \in \mathbb{N}\}.$$



$$\mathcal{R}(T') = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$
$$\mathcal{L}(T') = \{(ab)^n a | n \in \mathbb{N}\}.$$

Equivalence

$$\mathcal{R}(T) = \mathcal{R}(T')$$

Undecidable¹

Origin Equivalence²

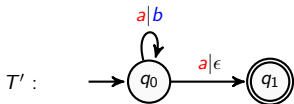
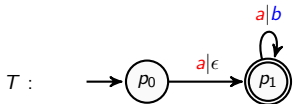
$$\mathcal{L}(T) = \mathcal{L}(T')$$

Decidable

¹Timothy V. Griffiths, 1968.

²Mikołaj Bojańczyk, 2014.

Inclusion and Equivalence Problems



$$\mathcal{R}(T) = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$

$$\mathcal{L}(T) = \{a(ab)^n | n \in \mathbb{N}\}.$$

$$\mathcal{R}(T') = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$

$$\mathcal{L}(T') = \{(ab)^n a | n \in \mathbb{N}\}.$$

Equivalence

?

Origin Equivalence ²

$$\mathcal{R}(T) = \mathcal{R}(T')$$

Undecidable ¹

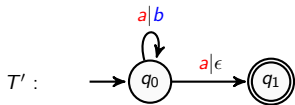
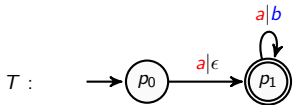
$$\mathcal{L}(T) = \mathcal{L}(T')$$

Decidable

¹Timothy V. Griffiths, 1968.

²Mikołaj Bojańczyk, 2014.

Inclusion and Equivalence Problems

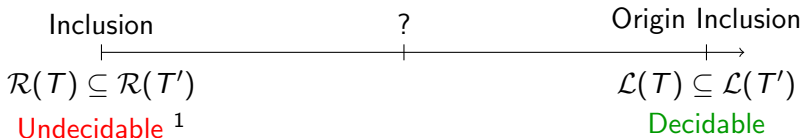


$$\mathcal{R}(T) = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$

$$\mathcal{L}(T) = \{a(ab)^n | n \in \mathbb{N}\}.$$

$$\mathcal{R}(T') = \{(a^{n+1}, b^n) | n \in \mathbb{N}\}.$$

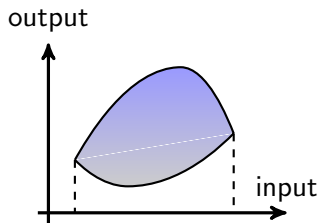
$$\mathcal{L}(T') = \{(ab)^n a | n \in \mathbb{N}\}.$$



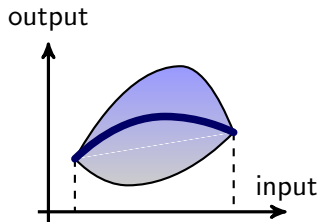
¹Timothy V. Griffiths, 1968.

²Mikołaj Bojańczyk, 2014.

Uniformisation Problems



Uniformisation Problems

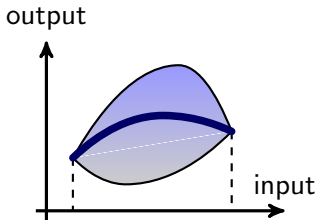


Uniformisation Problems

\mathcal{C} -Uniformisation

Given a transducer T , does there exist a transducer U such that

- $U \in \mathcal{C}$;
- $\mathcal{F}(U) \subseteq \mathcal{R}(T)$;
- $\text{dom}(\mathcal{F}(U)) = \text{dom}(\mathcal{R}(T))$?



Uniformisation Problems

\mathcal{C} -Uniformisation

Given a transducer T , does there exist a transducer U such that

- $U \in \mathcal{C}$;
 - $\mathcal{F}(U) \subseteq \mathcal{R}(T)$;
 - $\text{dom}(\mathcal{F}(U)) = \text{dom}(\mathcal{R}(T))$?
-
- $\mathcal{C} = \text{fRAT}$,

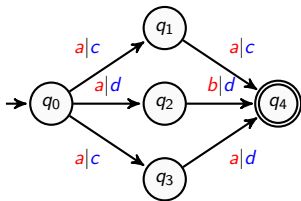
Uniformisation Problems

\mathcal{C} -Uniformisation

Given a transducer T , does there exist a transducer U such that

- $U \in \mathcal{C}$;
- $\mathcal{F}(U) \subseteq \mathcal{R}(T)$;
- $\text{dom}(\mathcal{F}(U)) = \text{dom}(\mathcal{R}(T))$?

- $\mathcal{C} = \text{fRAT}$,



$$\mathcal{R}(T) = \{(aa, cc), (ab, dd), (aa, cd)\}.$$

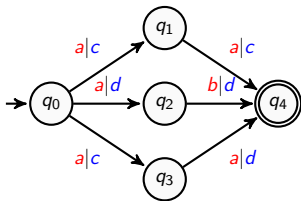
Uniformisation Problems

\mathcal{C} -Uniformisation

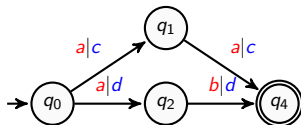
Given a transducer T , does there exist a transducer U such that

- $U \in \mathcal{C}$;
- $\mathcal{F}(U) \subseteq \mathcal{R}(T)$;
- $\text{dom}(\mathcal{F}(U)) = \text{dom}(\mathcal{R}(T))$?

- $\mathcal{C} = \text{fRAT}$,



$$\mathcal{R}(T) = \{(aa, cc), (ab, dd), (aa, cd)\}.$$



$$\mathcal{R}(U) = \{(aa, cc), (ab, dd)\}.$$

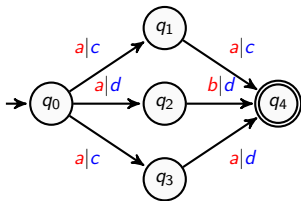
Uniformisation Problems

\mathcal{C} -Uniformisation

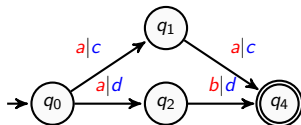
Given a transducer T , does there exist a transducer U such that

- $U \in \mathcal{C}$;
- $\mathcal{F}(U) \subseteq \mathcal{R}(T)$;
- $\text{dom}(\mathcal{F}(U)) = \text{dom}(\mathcal{R}(T))$?

- $\mathcal{C} = \text{fRAT}$, **decidable**. [Kojiro Kobayashi, 1969]



$$\mathcal{R}(T) = \{(aa, cc), (ab, dd), (aa, cd)\}.$$



$$\mathcal{R}(U) = \{(aa, cc), (ab, dd)\}.$$

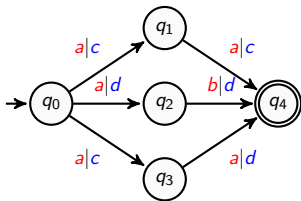
Uniformisation Problems

\mathcal{C} -Uniformisation

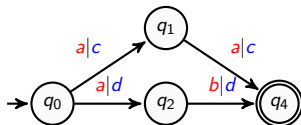
Given a transducer T , does there exist a transducer U such that

- $U \in \mathcal{C}$;
- $\mathcal{F}(U) \subseteq \mathcal{R}(T)$;
- $\text{dom}(\mathcal{F}(U)) = \text{dom}(\mathcal{R}(T))$?

- $\mathcal{C} = \text{fRAT}$, **decidable**. [Kojiro Kobayashi, 1969]
- $\mathcal{C} = \text{SEQ}$,



$$\mathcal{R}(T) = \{(aa, cc), (ab, dd), (aa, cd)\}.$$



$$\mathcal{R}(U) = \{(aa, cc), (ab, dd)\}.$$

Uniformisation Problems

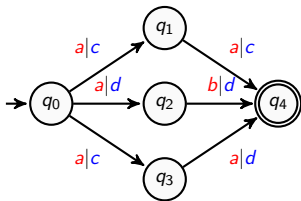
\mathcal{C} -Uniformisation

Given a transducer T , does there exist a transducer U such that

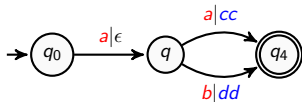
- $U \in \mathcal{C}$;
- $\mathcal{F}(U) \subseteq \mathcal{R}(T)$;
- $\text{dom}(\mathcal{F}(U)) = \text{dom}(\mathcal{R}(T))$?

- $\mathcal{C} = \text{FRAT}$, **decidable**. [Kojiro Kobayashi, 1969]

- $\mathcal{C} = \text{SEQ}$,



$$\mathcal{R}(T) = \{(aa, cc), (ab, dd), (aa, cd)\}.$$



$$\mathcal{R}(U') = \{(aa, cc), (ab, dd)\}.$$

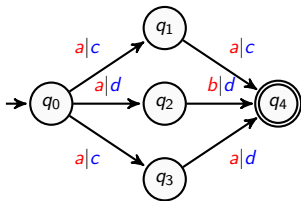
Uniformisation Problems

\mathcal{C} -Uniformisation

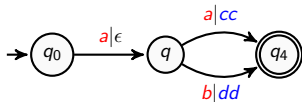
Given a transducer T , does there exist a transducer U such that

- $U \in \mathcal{C}$;
- $\mathcal{F}(U) \subseteq \mathcal{R}(T)$;
- $\text{dom}(\mathcal{F}(U)) = \text{dom}(\mathcal{R}(T))$?

- $\mathcal{C} = \text{fRAT}$, **decidable**. [Kojiro Kobayashi, 1969]
- $\mathcal{C} = \text{SEQ}$, **undecidable**. [Arnaud Carayol and Christof Löding, 2014]



$$\mathcal{R}(T) = \{(aa, cc), (ab, dd), (aa, cd)\}.$$



$$\mathcal{R}(U') = \{(aa, cc), (ab, dd)\}.$$

Uniformisation Problems

\mathcal{C} -Uniformisation

Given a transducer T , does there exist a transducer U such that

- $U \in \mathcal{C}$;
- $\mathcal{F}(U) \subseteq \mathcal{R}(T)$;
- $\text{dom}(\mathcal{F}(U)) = \text{dom}(\mathcal{R}(T))$?

SEQ-Unif.

$\mathcal{R}(U) \subseteq \mathcal{R}(T)$

Undecidable



Uniformisation Problems

\mathcal{C} -Uniformisation

Given a transducer T , does there exist a transducer U such that

- $U \in \mathcal{C}$;
- $\mathcal{F}(U) \subseteq \mathcal{R}(T)$;
- $\text{dom}(\mathcal{F}(U)) = \text{dom}(\mathcal{R}(T))$?

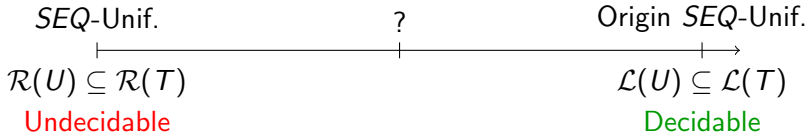


Uniformisation Problems

\mathcal{C} -Uniformisation

Given a transducer T , does there exist a transducer U such that

- $U \in \mathcal{C}$;
- $\mathcal{F}(U) \subseteq \mathcal{R}(T)$;
- $\text{dom}(\mathcal{F}(U)) = \text{dom}(\mathcal{R}(T))$?

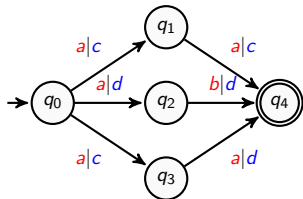


Uniformisation Problems

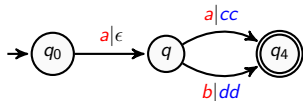
\mathcal{C} -Uniformisation

Given a transducer T , does there exist a transducer U such that

- $U \in \mathcal{C}$;
- $\mathcal{F}(U) \subseteq \mathcal{R}(T)$;
- $\text{dom}(\mathcal{F}(U)) = \text{dom}(\mathcal{R}(T))$?



$$\mathcal{R}(T) = \{(aa, cc), (ab, dd), (aa, cd)\},$$
$$\mathcal{L}(T) = \{acac, adbd, acad\}.$$



$$\mathcal{R}(U') = \{(aa, aa), (ab, bb)\},$$
$$\mathcal{L}(U') = \{aacc, abdd\}.$$

Resynchronisers

$w = a c b c a c d d$

Resynchronisers

$w = a c b c a c d d \mapsto (aba, ccdd)$

Resynchronisers

$w = a c b c a c d d \mapsto (aba, ccdd)$

$w' = a b c c c d d a \mapsto (aba, ccdd)$

Resynchronisers

$$w = a c b c a c d d \longmapsto (aba, cccdd)$$

$$w' = a b c c c d d a \longmapsto (aba, cccdd)$$

$w \sim w'$: w and w' have identical projections over Σ and Σ .

Resynchronisers

$$w = a c b c a c d d \longmapsto (aba, ccdd)$$

$$w' = a b c c c d d a \longmapsto (aba, ccdd)$$

$w \sim w'$: w and w' have identical projections over Σ and Σ .

Resynchroniser

A resynchroniser is a relation $\mathbb{S} \subset (\Sigma \cup \Sigma)^* \times (\Sigma \cup \Sigma)^*$ that preserves the input and output projections: $(w, w') \in \mathbb{S} \Rightarrow w \sim w'$.

Resynchronisers

$$w = a c b c a c d d \longmapsto (aba, cccdd)$$

$$w' = a b c c c d d a \longmapsto (aba, cccdd)$$

$w \sim w'$: w and w' have identical projections over Σ and Σ .

Resynchroniser

A resynchroniser is a relation $\mathbb{S} \subset (\Sigma \cup \Sigma)^* \times (\Sigma \cup \Sigma)^*$ that preserves the input and output projections: $(w, w') \in \mathbb{S} \Rightarrow w \sim w'$.

Identity resynchroniser: $\mathbb{I} = \{(w, w') \mid w = w'\}$,

Resynchronisers

$$w = a c b c a c d d \longmapsto (aba, cccdd)$$

$$w' = a b c c c d d a \longmapsto (aba, cccdd)$$

$w \sim w'$: w and w' have identical projections over Σ and Σ .

Resynchroniser

A resynchroniser is a relation $\mathbb{S} \subset (\Sigma \cup \Sigma)^* \times (\Sigma \cup \Sigma)^*$ that preserves the input and output projections: $(w, w') \in \mathbb{S} \Rightarrow w \sim w'$.

Identity resynchroniser: $\mathbb{I} = \{(w, w') \mid w = w'\},$

Universal resynchroniser: $\mathbb{U} = \{(w, w') \mid w \sim w'\},$

Resynchronisers

$$w = a c b c a c d d \longmapsto (aba, cccdd)$$

$$w' = a b c c c d d a \longmapsto (aba, cccdd)$$

$w \sim w'$: w and w' have identical projections over Σ and Σ .

Resynchroniser

A resynchroniser is a relation $\mathbb{S} \subset (\Sigma \cup \Sigma)^* \times (\Sigma \cup \Sigma)^*$ that preserves the input and output projections: $(w, w') \in \mathbb{S} \Rightarrow w \sim w'$.

$$\text{Identity resynchroniser: } \mathbb{I} = \{(w, w') \mid w = w'\},$$

$$\text{Universal resynchroniser: } \mathbb{U} = \{(w, w') \mid w \sim w'\},$$

$$k\text{-delay resynchroniser: } \mathbb{D}_k = \{(w, w') \mid d(w, w') \leq k\}.$$

Resynchronisers

$$\begin{array}{l} w = \overset{0}{a} \ c \ b \ c \ a \ c \ d \ d \longmapsto (aba, cccdd) \\ \quad | \\ w' = a \ b \ c \ c \ c \ d \ d \ a \longmapsto (aba, cccdd) \end{array}$$

$w \sim w'$: w and w' have identical projections over Σ and Σ .

Resynchroniser

A resynchroniser is a relation $\mathbb{S} \subset (\Sigma \cup \Sigma)^* \times (\Sigma \cup \Sigma)^*$ that preserves the input and output projections: $(w, w') \in \mathbb{S} \Rightarrow w \sim w'$.

Identity resynchroniser: $\mathbb{I} = \{(w, w') \mid w = w'\},$

Universal resynchroniser: $\mathbb{U} = \{(w, w') \mid w \sim w'\},$

k -delay resynchroniser: $\mathbb{D}_k = \{(w, w') \mid d(w, w') \leq k\}.$

Resynchronisers

$$\begin{array}{l} w = \overset{0}{a} \ c \ \overset{1}{b} \ c \ a \ c \ d \ d \longmapsto (aba, cccdd) \\ \quad \quad \quad | \quad / \\ w' = a \ b \ c \ c \ c \ d \ d \ a \longmapsto (aba, cccdd) \end{array}$$

$w \sim w'$: w and w' have identical projections over Σ and Σ .

Resynchroniser

A resynchroniser is a relation $\mathbb{S} \subset (\Sigma \cup \Sigma)^* \times (\Sigma \cup \Sigma)^*$ that preserves the input and output projections: $(w, w') \in \mathbb{S} \Rightarrow w \sim w'$.

Identity resynchroniser: $\mathbb{I} = \{(w, w') \mid w = w'\}$,

Universal resynchroniser: $\mathbb{U} = \{(w, w') \mid w \sim w'\}$,

k -delay resynchroniser: $\mathbb{D}_k = \{(w, w') \mid d(w, w') \leq k\}$.

Resynchronisers

$$\begin{array}{cccccccc} w = & \overset{0}{a} & c & \overset{1}{b} & c & \overset{3}{a} & c & d & d & \longmapsto & (aba, cccdd) \\ & | & & / & & \backslash & & & & & \\ w' = & a & b & c & c & c & d & d & a & \longmapsto & (aba, cccdd) \end{array}$$

$w \sim w'$: w and w' have identical projections over Σ and Σ .

Resynchroniser

A resynchroniser is a relation $\mathbb{S} \subset (\Sigma \cup \Sigma)^* \times (\Sigma \cup \Sigma)^*$ that preserves the input and output projections: $(w, w') \in \mathbb{S} \Rightarrow w \sim w'$.

Identity resynchroniser: $\mathbb{I} = \{(w, w') \mid w = w'\}$,

Universal resynchroniser: $\mathbb{U} = \{(w, w') \mid w \sim w'\}$,

k -delay resynchroniser: $\mathbb{D}_k = \{(w, w') \mid d(w, w') \leq k\}$.

Resynchronisers

$$\begin{array}{cccccccc} w = & \overset{0}{a} & c & \overset{1}{b} & c & \overset{3}{a} & c & d & d & \longmapsto & (aba, cccdd) \\ & | & & / & & \backslash & & & & & \\ w' = & a & b & c & c & c & d & d & a & \longmapsto & (aba, cccdd) \end{array}$$

$$d(w, w') = 3$$

$w \sim w'$: w and w' have identical projections over Σ and Σ .

Resynchroniser

A resynchroniser is a relation $\mathbb{S} \subset (\Sigma \cup \Sigma)^* \times (\Sigma \cup \Sigma)^*$ that preserves the input and output projections: $(w, w') \in \mathbb{S} \Rightarrow w \sim w'$.

Identity resynchroniser: $\mathbb{I} = \{(w, w') \mid w = w'\}$,

Universal resynchroniser: $\mathbb{U} = \{(w, w') \mid w \sim w'\}$,

k -delay resynchroniser: $\mathbb{D}_k = \{(w, w') \mid d(w, w') \leq k\}$.

Decision Problems modulo Resynchronisers

S-Inclusion Problem, $T \subseteq_S T'$

Given two transducers T, T' , does $\mathcal{L}(T) \subseteq \mathbb{S}(\mathcal{L}(T'))$ hold ?

\mathbb{S} -Inclusion Problem, $T \subseteq_{\mathbb{S}} T'$

Given two transducers T, T' , does $\mathcal{L}(T) \subseteq \mathbb{S}(\mathcal{L}(T'))$ hold ?

- The \mathbb{I} -inclusion problem is **decidable** (PSPACE-C).

\mathbb{S} -Inclusion Problem, $T \subseteq_{\mathbb{S}} T'$

Given two transducers T, T' , does $\mathcal{L}(T) \subseteq \mathbb{S}(\mathcal{L}(T'))$ hold ?

- The \mathbb{I} -inclusion problem is **decidable** (PSPACE-C).

$$T \subseteq_{\mathbb{I}} T' \Leftrightarrow \mathcal{L}(T) \subseteq \mathcal{L}(T').$$

\mathbb{S} -Inclusion Problem, $T \subseteq_{\mathbb{S}} T'$

Given two transducers T, T' , does $\mathcal{L}(T) \subseteq \mathbb{S}(\mathcal{L}(T'))$ hold ?

- The \mathbb{I} -inclusion problem is **decidable** (PSPACE-C).

$$T \subseteq_{\mathbb{I}} T' \Leftrightarrow \mathcal{L}(T) \subseteq \mathcal{L}(T').$$

- The \mathbb{U} -inclusion problem is **undecidable**.

\mathbb{S} -Inclusion Problem, $T \subseteq_{\mathbb{S}} T'$

Given two transducers T, T' , does $\mathcal{L}(T) \subseteq \mathbb{S}(\mathcal{L}(T'))$ hold ?

- The \mathbb{I} -inclusion problem is **decidable** (PSPACE-C).

$$T \subseteq_{\mathbb{I}} T' \Leftrightarrow \mathcal{L}(T) \subseteq \mathcal{L}(T').$$

- The \mathbb{U} -inclusion problem is **undecidable**.

$$T \subseteq_{\mathbb{U}} T' \Leftrightarrow \mathcal{R}(T) \subseteq \mathcal{R}(T').$$

Decision Problems modulo Resynchronisers

\mathbb{S} -Inclusion Problem, $T \subseteq_{\mathbb{S}} T'$

Given two transducers T, T' , does $\mathcal{L}(T) \subseteq \mathbb{S}(\mathcal{L}(T'))$ hold ?

- The \mathbb{I} -inclusion problem is **decidable** (PSPACE-C).
 $T \subseteq_{\mathbb{I}} T' \Leftrightarrow \mathcal{L}(T) \subseteq \mathcal{L}(T')$.
- The \mathbb{U} -inclusion problem is **undecidable**.
 $T \subseteq_{\mathbb{U}} T' \Leftrightarrow \mathcal{R}(T) \subseteq \mathcal{R}(T')$.
- For all $k \in \mathbb{N}$, the \mathbb{D}_k -inc. problem is **decidable** (PSPACE-C).

Decision Problems modulo Resynchronisers

\mathbb{S} -Inclusion Problem, $T \subseteq_{\mathbb{S}} T'$

Given two transducers T, T' , does $\mathcal{L}(T) \subseteq \mathbb{S}(\mathcal{L}(T'))$ hold ?

- The \mathbb{I} -inclusion problem is **decidable** (PSPACE-C).
 $T \subseteq_{\mathbb{I}} T' \Leftrightarrow \mathcal{L}(T) \subseteq \mathcal{L}(T')$.
- The \mathbb{U} -inclusion problem is **undecidable**.
 $T \subseteq_{\mathbb{U}} T' \Leftrightarrow \mathcal{R}(T) \subseteq \mathcal{R}(T')$.
- For all $k \in \mathbb{N}$, the \mathbb{D}_k -inc. problem is **decidable** (PSPACE-C).
 $\exists T'_k$ s.t. $\mathcal{L}(T'_k) = \mathbb{D}_k(\mathcal{L}(T'))$. Then $T \subseteq_{\mathbb{D}_k} T' \Leftrightarrow T \subseteq_{\mathbb{I}} T'_k$.

\mathbb{S} -Inclusion Problem, $T \subseteq_{\mathbb{S}} T'$

Given two transducers T, T' , does $\mathcal{L}(T) \subseteq \mathbb{S}(\mathcal{L}(T'))$ hold ?

- The \mathbb{I} -inclusion problem is **decidable** (PSPACE-C).
- The \mathbb{U} -inclusion problem is **undecidable**.
- For all $k \in \mathbb{N}$, the \mathbb{D}_k -inc. problem is **decidable** (PSPACE-C).

Decision Problems modulo Resynchronisers

\mathbb{S} -Inclusion Problem, $T \subseteq_{\mathbb{S}} T'$

Given two transducers T, T' , does $\mathcal{L}(T) \subseteq \mathbb{S}(\mathcal{L}(T'))$ hold ?

- The \mathbb{I} -inclusion problem is **decidable** (PSPACE-C).
- The \mathbb{U} -inclusion problem is **undecidable**.
- For all $k \in \mathbb{N}$, the \mathbb{D}_k -inc. problem is **decidable** (PSPACE-C).

\mathbb{S} -Equivalence problem, $T \equiv_{\mathbb{S}} T'$

Do $T \subseteq_{\mathbb{S}} T'$ and $T' \subseteq_{\mathbb{S}} T$ hold ?

- The \mathbb{I} -equivalence problem is **decidable** (PSPACE-C).
- The \mathbb{U} -equivalence problem is **undecidable**.
- For all $k \in \mathbb{N}$, the \mathbb{D}_k -equ. problem is **decidable** (PSPACE-C).

S-SEQ-Uniformisation Problem

Given a transducer T , does there exist a transducer U such that

- U is sequential;
- $U \subseteq_S T$;
- $\text{dom}(\mathcal{F}(U)) = \text{dom}(\mathcal{R}(T))$?

S-SEQ-Uniformisation Problem

Given a transducer T , does there exist a transducer U such that

- U is sequential;
 - $U \subseteq_S T$;
 - $\text{dom}(\mathcal{F}(U)) = \text{dom}(\mathcal{R}(T))$?
-
- For all $k \in \mathbb{N}$, the \mathbb{D}_k -SEQ-unif. problem is **decidable** (EXPTIME-C).

S-SEQ-Uniformisation Problem

Given a transducer T , does there exist a transducer U such that

- U is sequential;
 - $U \subseteq_S T$;
 - $\text{dom}(\mathcal{F}(U)) = \text{dom}(\mathcal{R}(T))$?
-
- For all $k \in \mathbb{N}$, the \mathbb{D}_k -SEQ-unif. problem is **decidable** (EXPTIME-C).
Reduction to a two-players safety game.

Main Results : Finitely Ambiguous Transducers

Theorem

Completeness for all problems is achieved for finitely ambiguous transducers.

Main Results : Finitely Ambiguous Transducers

Theorem

Completeness for all problems is achieved for finitely ambiguous transducers.

- $\mathcal{R}(T) \subseteq \mathcal{R}(T') \Leftrightarrow T \subseteq_{\mathbb{D}_k} T'$ (for some computable k).
- $\mathcal{R}(T) = \mathcal{R}(T') \Leftrightarrow T \equiv_{\mathbb{D}_k} T'$ (for some computable k).
- T SEQ-unif. $\Leftrightarrow T$ \mathbb{D}_k -SEQ unif. (for some computable k).

Main Results : Finitely Ambiguous Transducers

Theorem

Completeness for all problems is achieved for finitely ambiguous transducers.

- $\mathcal{R}(T) \subseteq \mathcal{R}(T') \Leftrightarrow T \subseteq_{\mathbb{D}_k} T'$ (for some computable k).
- $\mathcal{R}(T) = \mathcal{R}(T') \Leftrightarrow T \equiv_{\mathbb{D}_k} T'$ (for some computable k).
- T SEQ-unif. $\Leftrightarrow T$ \mathbb{D}_k -SEQ unif. (for some computable k).

Main Results : Finitely Ambiguous Transducers

Theorem

Completeness for all problems is achieved for finitely ambiguous transducers.

- $\mathcal{R}(T) \subseteq \mathcal{R}(T') \Leftrightarrow T \subseteq_{\mathbb{D}_k} T'$ (for some computable k).
- $\mathcal{R}(T) = \mathcal{R}(T') \Leftrightarrow T \equiv_{\mathbb{D}_k} T'$ (for some computable k).
- T SEQ-unif. $\Leftrightarrow T$ \mathbb{D}_k -SEQ unif. (for some computable k).

Theorem

Inclusion, Equivalence and SEQ-Uniformisation are **decidable** for finitely ambiguous transducers.

Main Results : Finitely Ambiguous Transducers

Theorem

Completeness for all problems is achieved for finitely ambiguous transducers.

- $\mathcal{R}(T) \subseteq \mathcal{R}(T') \Leftrightarrow T \subseteq_{\mathbb{D}_k} T'$ (for some computable k).
- $\mathcal{R}(T) = \mathcal{R}(T') \Leftrightarrow T \equiv_{\mathbb{D}_k} T'$ (for some computable k).
- T SEQ-unif. $\Leftrightarrow T$ \mathbb{D}_k -SEQ unif. (for some computable k).

Theorem

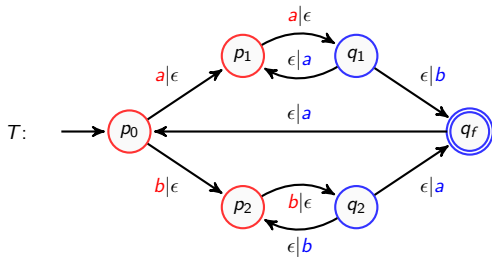
Inclusion, Equivalence and SEQ-Uniformisation are **decidable** for finitely ambiguous transducers.

Equivalence was already known to be decidable.

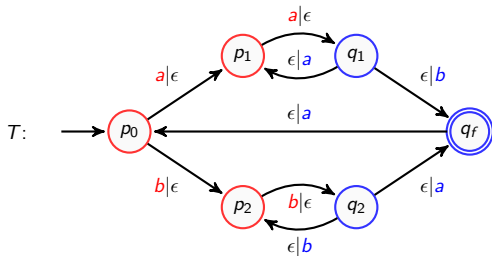
Eitan M. Gurari and Oscar H. Ibarra. A note on finite-valued and finitely ambiguous transducers. 1983.

Main Results : Deterministic Transducers

Main Results : Deterministic Transducers

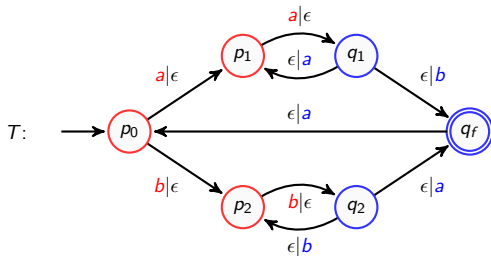


Main Results : Deterministic Transducers



- Inclusion is **undecidable** for deterministic transducers.
Patrick C. Fischer and Arnold L. Rosenberg. Multitape one-way nonwriting automata. 1968.
- Equivalence is **decidable** for deterministic transducers.
Malcolm Bird. The equivalence problem for deterministic two-tape automata. 1973.

Main Results : Deterministic Transducers



- Inclusion is **undecidable** for deterministic transducers.
Patrick C. Fischer and Arnold L. Rosenberg. Multitape one-way nonwriting automata. 1968.
- Equivalence is **decidable** for deterministic transducers.
Malcolm Bird. The equivalence problem for deterministic two-tape automata. 1973.

Theorem

SEQ -Uniformisation is **decidable** for deterministic transducers.

Conclusion

	Inclusion	Equivalence	SEQ-Unif.
<i>RAT</i>	UNDECIDABLE	UNDECIDABLE [1]	UNDECIDABLE [2]
<i>RAT</i> modulo \mathbb{S}			
Finite-valued	DECIDABLE	DECIDABLE [3]	
<i>DRAT</i>	UNDECIDABLE [4]	DECIDABLE [5]	

[1] Timothy V. Griffiths, 1968.

[2] Arnaud Carayol and Christof Löding, 2014.

[3] Eitan M. Gurari and Oscar H. Ibarra, 1983.

[4] Patrick C. Fischer and Arnold L. Rosenberg, 1968.

[5] Malcolm Bird, 1973.

Conclusion

	Inclusion	Equivalence	SEQ-Unif.
<i>RAT</i>	UNDECIDABLE	UNDECIDABLE [1]	UNDECIDABLE [2]
<i>RAT</i> modulo \mathbb{S}	PSPACE-C	PSPACE-C	EXPTIME-C
Finite-valued	DECIDABLE	DECIDABLE [3]	DECIDABLE
<i>DRAT</i>	UNDECIDABLE [4]	DECIDABLE [5]	DECIDABLE

- [1] Timothy V. Griffiths, 1968.
- [2] Arnaud Carayol and Christof Löding, 2014.
- [3] Eitan M. Gurari and Oscar H. Ibarra, 1983.
- [4] Patrick C. Fischer and Arnold L. Rosenberg, 1968.
- [5] Malcolm Bird, 1973.