Finite-valued Streaming String Transducers

Emmanuel Filiot, Ismaël Jecker, Christof Löding,
Anca Muscholl, Gabriele Puppis, Sarah Winter

Finite-valued regular relations are binary relations that enjoy good algorithmic properties

Finite-valued regular relations are binary relations that enjoy good algorithmic properties

Theorem: We can decide in polynomial space whether a given **SST** defines a **finite-valued regular relation**

Theorem: Every **finite-valued regular relation** can be decomposed into a finite union of **regular functions**

TRANSDUCERS

Transducers are abstract machines that recognise **relations**

$$\mathsf{R} \subseteq \Sigma^* \times \Gamma^*$$

Rational relations	recognised by finite state transducers
Regular relations	recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

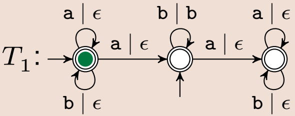
Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)

Input: a b b a a a b b b a b a a a

Output:



Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

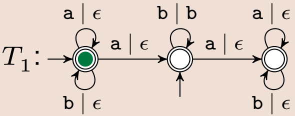
Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)

Input: a b b a a a b b b a b a a a

Output:



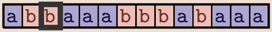
Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

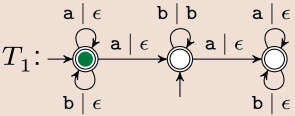
Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)

Input: 

Output:



Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

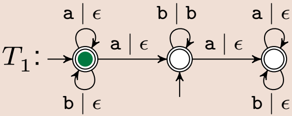
Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)

Input: a b b a a a b b b a b a a a

Output:



Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

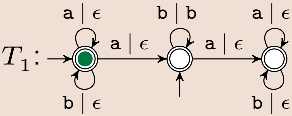
Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)

Input: a b b a a a b b b a b a a a

Output:



Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

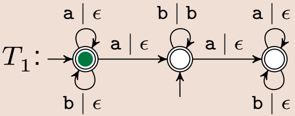
TRANSDUCERS

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)

Input: a b b a a a b b b a b a a a

Output:



Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

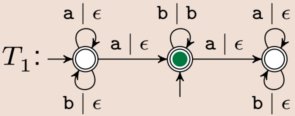
Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)

Input: a b b a a a b b b a b a a a

Output:



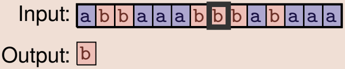
Rational relations recognised by finite state transducers

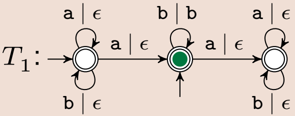
Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)





Rational relations recognised by finite state transducers

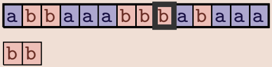
Regular relations recognised by streaming string transducers

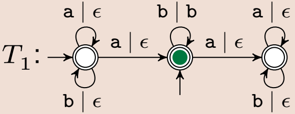
Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)

Input:
Output:





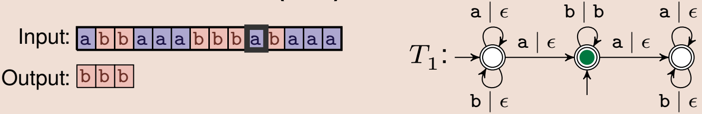
Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)



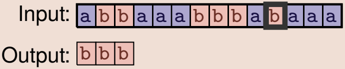
Rational relations recognised by finite state transducers

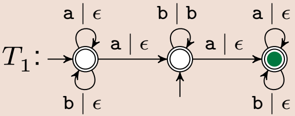
Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)





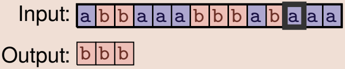
Rational relations recognised by finite state transducers

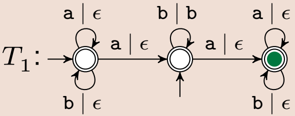
Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)





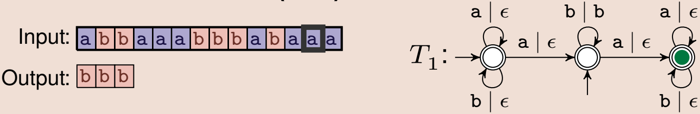
Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)



Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

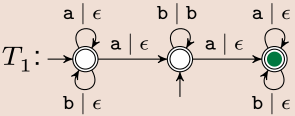
Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)

Input: a b b a a a b b b a b a a a

Output: b b b



Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

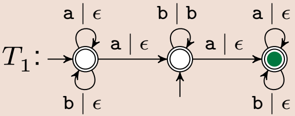
Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)

Input: a b b a a a b b b a b a a a

Output: b b b



Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

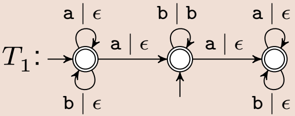
Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)

Input: a b b a a a b b b a b a a a

Output: b b b



T_1 produces one of the b-blocks of its input

Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

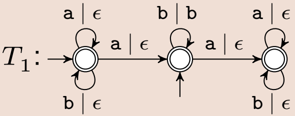
Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)

Input: a b b a a a b b b a b a a a

Output: b b b   b   b b



T_1 produces one of the b-blocks of its input

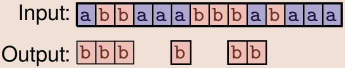
Rational relations recognised by finite state transducers

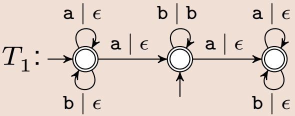
Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)





T_1 produces one of the b-blocks of its input

Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

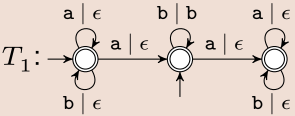
Transducers

Transducers are abstract machines that recognise **relations**

Finite state transducer (FST)

Input: abbaaabbbabaaa

Output: bbb b bb



T_1 produces one of the b-blocks of its input

Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

Transducers

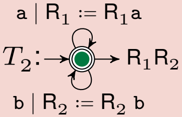
Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)

Input: a b b a a a b b b a b a a a

R_1:

R_2:



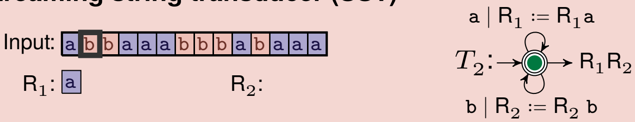
Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

TRANSDUCERS

Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)



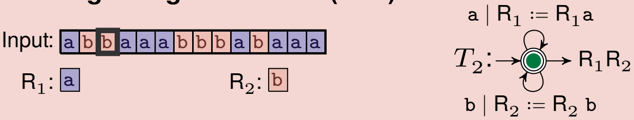
Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)

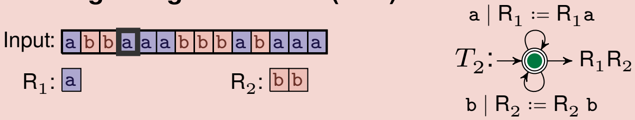


Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)



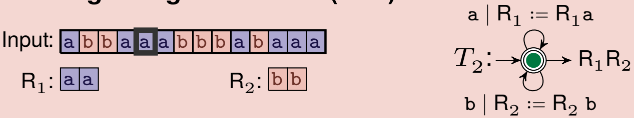
Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)



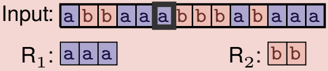
Rational relations recognised by finite state transducers

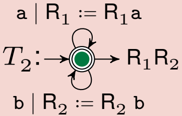
Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)





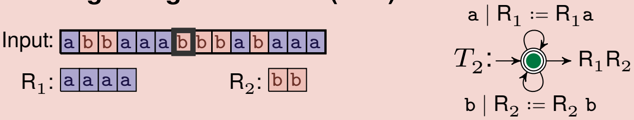
Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)



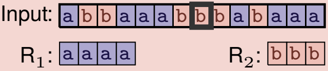
Rational relations recognised by finite state transducers

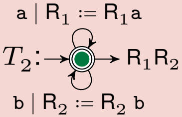
Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)





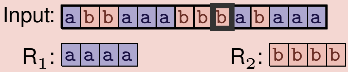
Rational relations recognised by finite state transducers

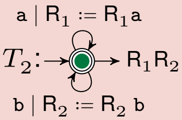
Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)





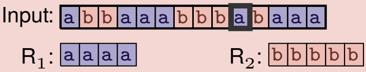
Rational relations recognised by finite state transducers

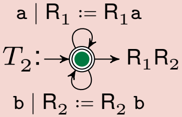
Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)





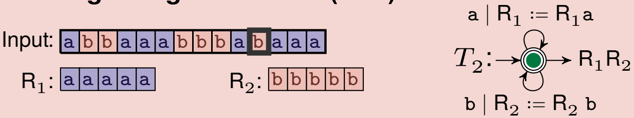
Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)



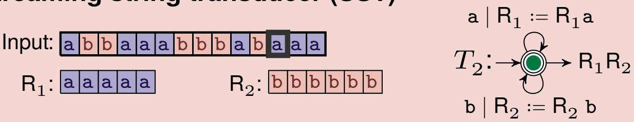
Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)



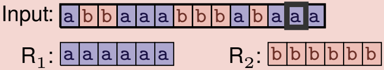
Rational relations recognised by finite state transducers

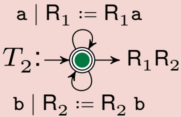
Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)





Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

TRANSDUCERS

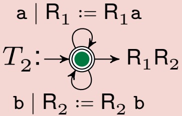
Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)

Input: a b b a a a b b b a b a a a

R_1: a a a a a a a

R_2: b b b b b b



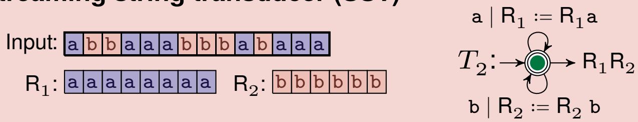
Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

Transducers

Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)



Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

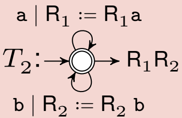
Transducers

Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)

Input: a b b a a a b b b a b a a a

Output: a a a a a a a a b b b b b b



Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

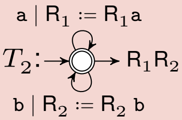
Transducers

Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)

Input: a b b a a a b b b a b a a a

Output: a a a a a a a a b b b b b b



Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

Transducers

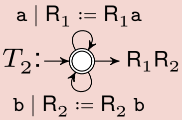
Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)

Input: a b b a a a b b b a b a a a

Output: a a a a a a a a b b b b b b

T_2 sorts its input



Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

Transducers

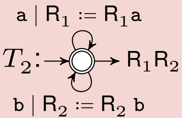
Transducers are abstract machines that recognise **relations**

Streaming string transducer (SST)

Input: a b b a a a b b b a b a a a

Output: a a a a a a a a b b b b b b

T_2 sorts its input



Rational relations recognised by finite state transducers

Regular relations recognised by streaming string transducers

Transducers

Rational relations	recognised by finite state transducers
Regular relations	recognised by streaming string transducers

Transducers

Equivalence of **rational relations** is **undecidable**

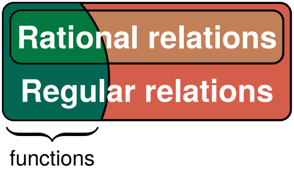
Rational relations	recognised by finite state transducers
Regular relations	recognised by streaming string transducers

TRANSDUCERS

Equivalence of **rational relations** is **undecidable**

Equivalence of **regular functions** is **decidable**

↳ each input is mapped to at most 1 output



recognised by finite state transducers

recognised by streaming string transducers

Transducers

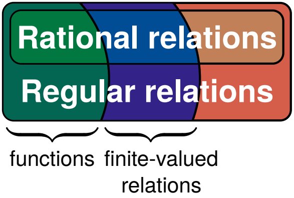
Equivalence of **rational relations** is **undecidable**

Equivalence of **regular functions** is **decidable**

↳ each input is mapped to at most 1 output

Equivalence of **finite-valued regular relations** is **decidable**

↳ $\exists \mathbf{k} \in \mathbb{N}$ s.t. each input is mapped to at most **k** outputs



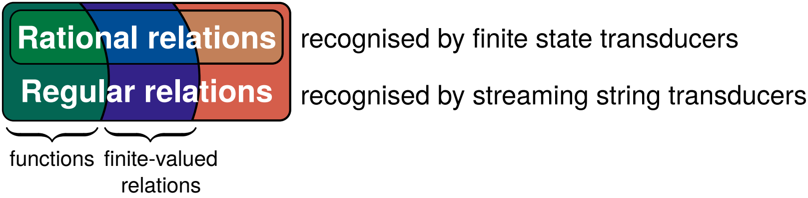
recognised by finite state transducers

recognised by streaming string transducers

Transducers

Theorem: We can decide in polynomial space whether a given **SST** defines a **finite-valued regular relation**

Theorem: Every **finite-valued regular relation** can be decomposed into a finite union of **regular functions**



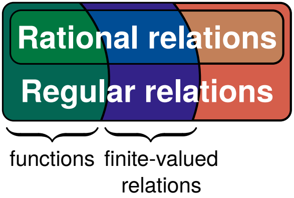
TRANSDUCERS

Theorem: We can decide in polynomial space whether a given **SST** defines a **finite-valued regular relation**

Theorem: Every **finite-valued regular relation** can be decomposed into a finite union of **regular functions**

⇒ The equivalence problem for finite-valued SST is in ELEMENTARY

⇒ Finite-valued **2-way FST** are as expressive as finite-valued SST



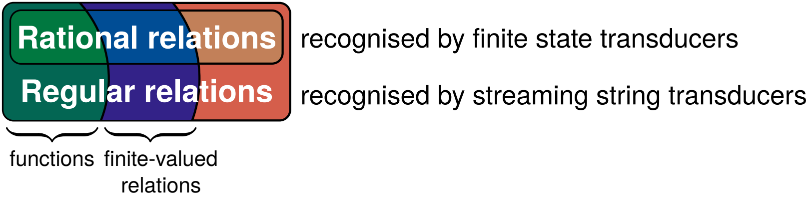
recognised by finite state transducers

recognised by streaming string transducers

Transducers

Theorem: We can decide in polynomial space whether a given SST defines a **finite-valued regular relation**

Theorem: Every **finite-valued regular relation** can be decomposed into a finite union of **regular functions**



Transducers

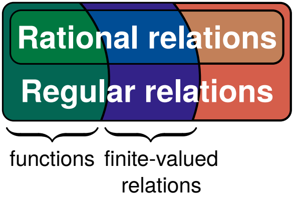
Theorem: We can decide in polynomial space whether a given **SST** defines a **finite-valued regular relation**

Theorem: Every **finite-valued regular relation** can be decomposed into a finite union of **regular functions**

Raised as open problems by [2011. Alur, Deshmukh]

Known to hold for FST [1989. Weber], [1993. Weber]

...and for SST with a single register [2017. Gallot et al.]



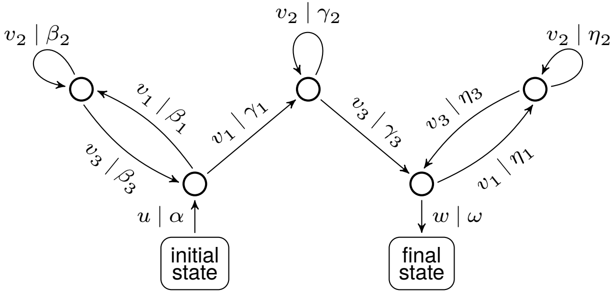
recognised by finite state transducers

recognised by streaming string transducers

Deciding Finite Valuedness

Theorem: We can decide in polynomial space whether a given SST defines a **finite-valued regular relation**

Forbidden pattern: (inspired by [2008. De Souza])



- The relation recognised by the pattern is not 1-valued
- The substitutions produced on the loops have idempotent structure

The Decomposition Theorem

Theorem: Every **finite-valued regular relation** can be decomposed into a finite union of **regular functions**

Proof: (inspired by [2008. Sakarovitch, de Souza] and relying on [2023. FJLW])

The Decomposition Theorem

Theorem: Every **finite-valued regular relation** can be decomposed into a finite union of **regular functions**

Proof: (inspired by [2008. Sakarovitch, de Souza] and relying on [2023. FJLW])

$\mathcal{T}$ that associates at most **k** outputs to each input

$\mathcal{T}_1$, ..., $\mathcal{T}_k$ that all associate at most **1** output to each input

The Decomposition Theorem

Theorem: Every **finite-valued regular relation** can be decomposed into a finite union of **regular functions**

Proof: (inspired by [2008. Sakarovitch, de Souza] and relying on [2023. FJLW])

$\mathcal{T}$ that associates at most **k** outputs to each input

$\mathcal{T}$' that has at most **k** runs on each input

$\mathcal{T}'_1$, ..., $\mathcal{T}'_k$ that all have at most **1** run on each input

$\mathcal{T}_1$, ..., $\mathcal{T}_k$ that all associate at most **1** output to each input

THE DECOMPOSITION THEOREM

Theorem: Every **finite-valued regular relation** can be decomposed into a finite union of **regular functions**

Proof: (inspired by [2008. Sakarovitch, de Souza] and relying on [2023. FJLW])

$\mathcal{T}$ that associates at most **k** outputs to each input

$\mathcal{T}$' that has at most **k** runs on each input

$\mathcal{T}'_i$ copies the ith lexicographically smallest run of $\mathcal{T}'$

$\mathcal{T}'_1$, ..., $\mathcal{T}'_k$ that all have at most **1** run on each input

$\mathcal{T}_1$, ..., $\mathcal{T}_k$ that all associate at most **1** output to each input

The Decomposition Theorem

Theorem: Every **finite-valued regular relation** can be decomposed into a finite union of **regular functions**

Proof: (inspired by [2008. Sakarovitch, de Souza] and relying on [2023. FJLW])

$\mathcal{T}$ that associates at most **k** outputs to each input

$\mathcal{T}$' that has at most **k** runs on each input

⇓

$\mathcal{T}'_1$, ..., $\mathcal{T}'_k$ that all have at most **1** run on each input

$\mathcal{T}_1$, ..., $\mathcal{T}_k$ that all associate at most **1** output to each input

The Decomposition Theorem

Theorem: Every **finite-valued regular relation** can be decomposed into a finite union of **regular functions**

Proof: (inspired by [2008. Sakarovitch, de Souza] and relying on [2023. FJLW])

$\mathcal{T}$ that associates at most **k** outputs to each input

$\mathcal{T}$' that has at most **k** runs on each input

$\Downarrow$

$\mathcal{T}'_1$, ..., $\mathcal{T}'_k$ that all have at most **1** run on each input

$\Downarrow$

$\mathcal{T}_1$, ..., $\mathcal{T}_k$ that all associate at most **1** output to each input

The Decomposition Theorem

Theorem: Every **finite-valued regular relation** can be decomposed into a finite union of **regular functions**

Proof: (inspired by [2008. Sakarovitch, de Souza] and relying on [2023. FJLW])

$\mathcal{T}$ that associates at most **k** outputs to each input

⇓ $\mathcal{T}'$ only keeps the runs of $\mathcal{T}$ that are **far** from each other

$\mathcal{T}$' that has at most **k** runs on each input

⇓

$\mathcal{T}'_1$, ..., $\mathcal{T}'_k$ that all have at most **1** run on each input

⇓

$\mathcal{T}_1$, ..., $\mathcal{T}_k$ that all associate at most **1** output to each input

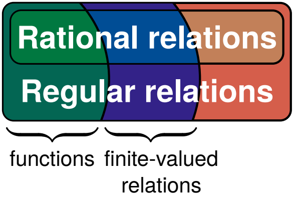
Transducers

Theorem: We can decide in polynomial space whether a given **SST** defines a **finite-valued regular relation**

Theorem: Every **finite-valued regular relation** can be decomposed into a finite union of **regular functions**

$\Rightarrow$ The equivalence problem for finite-valued SST is in Elementary

$\Rightarrow$ Finite-valued **2-way FST** are as expressive as finite-valued SST



recognised by finite state transducers

recognised by streaming string transducers