From Two-Way to One-Way Finite State Transducers

Emmanuel Filiot ULB-FNRS
Olivier Gauwin UBordeaux (LaBRI)
Pierre-Alain Reynier UMarseille (LIF)
Frédéric Servais UHasselt

Highlights of Logic, Games and Automata, 2013

Overview

- finite state automata: 1-way $\equiv$ 2-way
  [Rabin Scott 59][Shepherdson 59]

Overview

- finite state automata: 1-way $\equiv$ 2-way
  [Rabin Scott 59][Shepherdson 59]
- finite state (string) transducers: 1-way $<$ 2-way

Overview

- finite state automata: 1-way $\equiv$ 2-way
  [Rabin Scott 59][Shepherdson 59]
- finite state (string) transducers: 1-way $<$ 2-way
- This paper: 1-way definability of 2-way transducers
- Result: decidable for functions
- How? extend Rabin and Scott's proof to transducers

Two-Way Automata

A non-deterministic two-way finite state automaton (*2NFA*) is

$$A = (Q, q_0, F, \Delta)$$

where

$$\Delta \subseteq Q \times \Sigma \times Q \times \{+1, -1\}$$

Two-Way Automata

A non-deterministic two-way finite state automaton (*2NFA*) is

$$A = (Q, q_0, F, \Delta)$$

where

$$\Delta \subseteq Q \times \Sigma \times Q \times \{+1, -1\}$$

Theorem (Rabin and Scott 59, Shepherdson 59, Büchi)

2NFA = *2DFA* = *1NFA* = *1DFA* = *MSO*

One and Two-Way Transducers

A non-deterministic two-way finite state transducer (*2NFT*) is

$$T = (Q, q_0, F, \Delta)$$

where

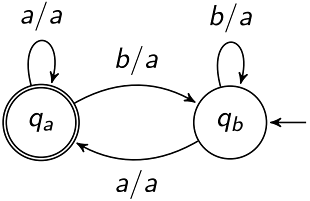
$$\Delta \subseteq Q \times \Sigma \times Q \times \Sigma^* \times \{+1, -1\}$$

- an output of a word = concatenation of the outputs of each transition of a successful run
- non-determinism $\implies$ string-to-string relation
- functional transducer $=_{def}$ defines a function
- one-way $=_{def}$ no transition with -1

Examples

$\Sigma = \{a, b\}$

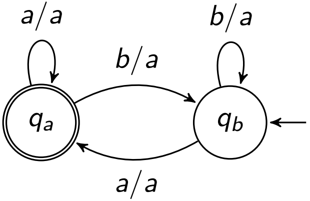
1. *1DFT*: If the last letter is 'a', replace all letters by 'a', otherwise undefined.



Examples

$\Sigma = \{a, b\}$

1. *1DFT*: If the last letter is 'a', replace all letters by 'a', otherwise undefined.

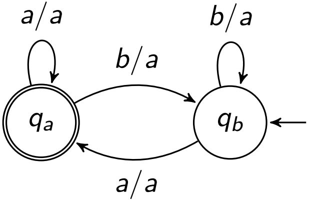


2. *1NFT*: If the last letter is 'a' (resp. 'b'), replace all letters by 'a' (resp. 'b').

Examples

$\Sigma = \{a, b\}$

1. *1DFT*: If the last letter is 'a', replace all letters by 'a', otherwise undefined.

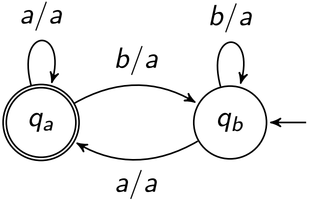


2. *1NFT*: If the last letter is 'a' (resp. 'b'), replace all letters by 'a' (resp. 'b').
   - *1DFT* $<$ *1NFT* (even functional)
   - It is definable by a deterministic 2-way transducer !

Examples

$\Sigma = \{a, b\}$

1. *1DFT*: If the last letter is 'a', replace all letters by 'a', otherwise undefined.



2. *1NFT*: If the last letter is 'a' (resp. 'b'), replace all letters by 'a' (resp. 'b').
    - *1DFT* $<$ *1NFT* (even functional)
    - It is definable by a deterministic 2-way transducer !
3. *2DFT*: $\#u\# \mapsto mirror(u)$ (Undefinable by a 1NFT)

Classes of Transductions

D="(input) deterministic"

f="functional"

1DFTs	*f1NFTs*	*1NFTs*
2DFTs	*f2NFTs*	*2NFTs*

Classes of Transductions

D="(input) deterministic"

f="functional"

$\#u\# \mapsto \text{mirror}(u)$

1DFTs	*f1NFTs*	*1NFTs*
$\cap$	$\cap$	$\cap$
2DFTs	*f2NFTs*	*2NFTs*

Classes of Transductions

D="(input) deterministic"

f="functional"

$$\{(a,a),(a,b)\}$$

$$\begin{array}{ccccc}
\textit{1DFTs} & & \textit{f1NFTs} & \subset & \textit{1NFTs} \\
\cap & & \cap & & \cap \\
\textit{2DFTs} & & \textit{f2NFTs} & \subset & \textit{2NFTs}
\end{array}$$

Classes of Transductions

D="(input) deterministic"

f="functional"

$$u \mapsto \begin{cases} a^{|u|} \text{ if } \mathit{last}(u) = a \\ u \text{ otherwise} \end{cases}$$

1DFTs	$\subsetneq$	*f1NFTs*	$\subsetneq$	*1NFTs*
$\cap$		$\cap$		$\cap$
2DFTs		*f2NFTs*	$\subsetneq$	*2NFTs*

Classes of Transductions

D="(input) deterministic"

f="functional"

$$\begin{array}{ccccc} \mathit{1DFTs} & \subset & \mathit{f1NFTs} & \subset & \mathit{1NFTs} \\ \cap & & \cap & & \cap \\ \mathit{2DFTs} & \equiv & \mathit{f2NFTs} & \subset & \mathit{2NFTs} \end{array}$$

[De Souza (13)]

$\equiv$ *MSOT* [Engelfriet,Hoogeboom (01)]

$\equiv$ Streaming String Transducers [Alur, Černý, 2010]

Classes of Transductions

D="(input) deterministic"

f="functional"

PTIME
[Choffrut (77)]
[Weber, Klemm (95)]
[Beal,Carton,Prieur,Sakarovitch(03)]
?

1DFTs	$\subset$	*f1NFTs*	$\subset$	*1NFTs*
$\cap$		$\cap$		$\cap$
2DFTs	$\equiv$	*f2NFTs*	$\subset$	*2NFTs*

[De Souza (13)]

$\equiv$ *MSOT* [Engelfriet,Hoogeboom (01)]

$\equiv$ Streaming String Transducers [Alur, Černý, 2010]

Classes of Transductions

D="(input) deterministic"

f="functional"

PTIME
[Choffrut (77)]
[Weber, Klemm (95)]
[Beal,Carton,Prieur,Sakarovitch(03)]
?

PTIME
[Schützenberger (75)]
[Gurari, Ibarra (83)]
[Beal,Carton,Prieur,Sakarovitch(03)]
?

1DFTs $\subseteq$ *f1NFTs* $\subseteq$ *1NFTs*

$\cap$ $\cap$ $\cap$

2DFTs $\equiv$ *f2NFTs* $\subseteq$ *2NFTs*

[De Souza (13)]

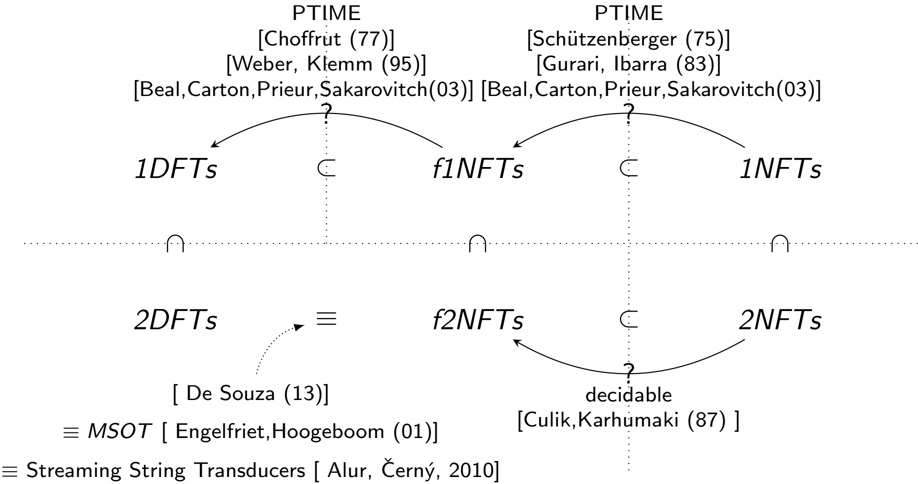
$\equiv$ *MSOT* [Engelfriet,Hoogeboom (01)]

$\equiv$ Streaming String Transducers [Alur, Černý, 2010]

Classes of Transductions

D="(input) deterministic"

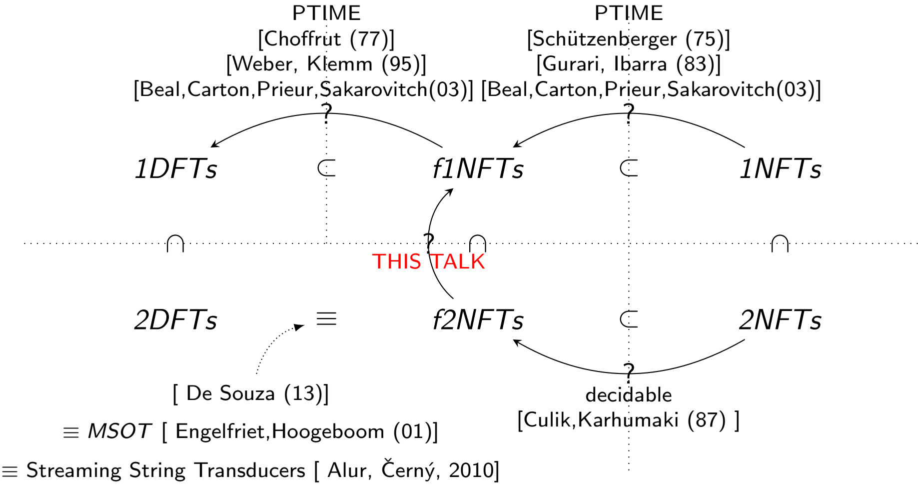
f="functional"



Classes of Transductions

D="(input) deterministic"

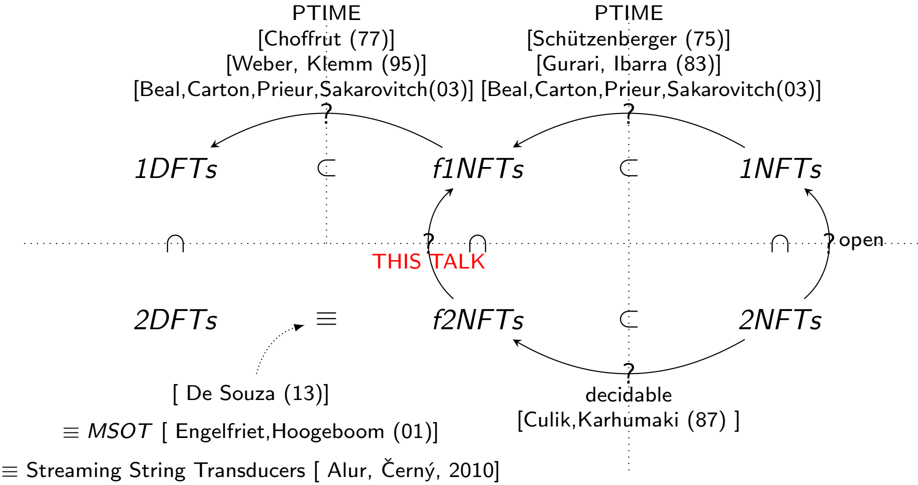
f="functional"



Classes of Transductions

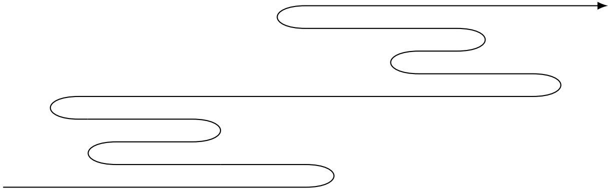
D="(input) deterministic"

f="functional"



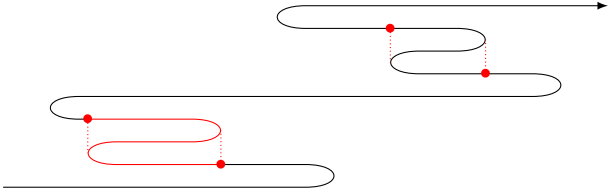
Rabin and Scott's proof for 2-Automata

- a run is made of many zigzags (moves of the input head)



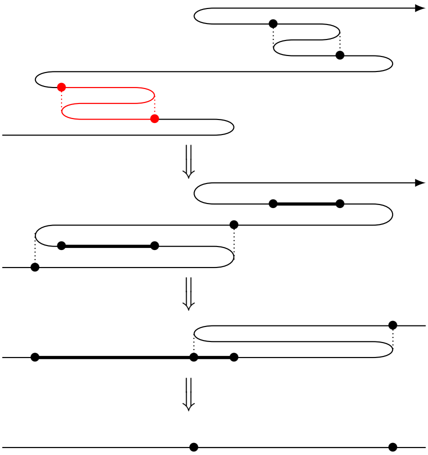
Rabin and Scott's proof for 2-Automata

- a run is made of many zigzags (moves of the input head)

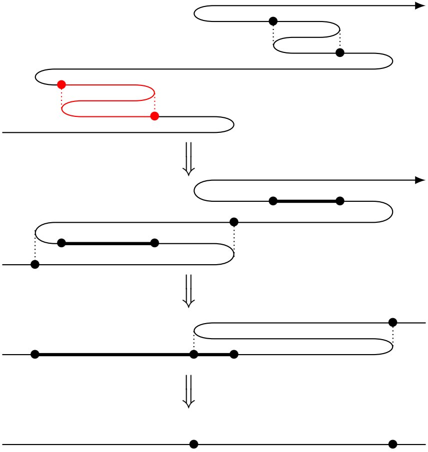


- A *z-motion* is an elementary zigzag.

Rabin and Scott's Proof: z-motions removal

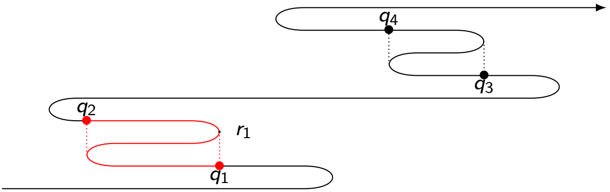


Rabin and Scott's Proof: z-motions removal



- **Def**: A shape is k crossing if any position is visited at most k times.
- **Thm**: Any k-crossing shape can be reduced to a line in k^2 steps.

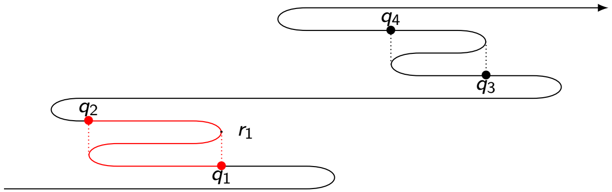
squeeze(A)



$S = \text{squeeze}(A)$ removes some z-motions of A.

1. simulates A
2. non-deterministically guesses that a z-motion starts (e.g. from q_1 to q_2)
3. **checks** that is indeed a z-motion and **simulates** it one-way
4. goes back to mode 1

squeeze(A)

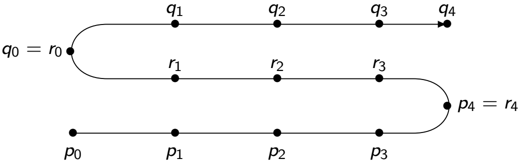


$S = \mathrm{squeeze}(A)$ removes some z-motions of A.

Iterate squeeze(A)

- Every accepted word has a one-way run in $\mathrm{squeeze}^{|Q|^2}(A)$
  $\Longrightarrow$ remove backward transitions to obtain a 1NFA equivalent to A.

How to simulate a z-motion run in one-way ?



Simulate the three passes in parallel ! (with triple of states)

Extension to transducers

- same canvas (Rabin and Scott)
- removal of z-motion:
  ➜ translate a z-motion transducer into a *f1NFT*
- not always possible $\rightarrow$ decision procedure

Extension to transducers

- same canvas (Rabin and Scott)
- removal of z-motion:
  ➜ translate a z-motion transducer into a *f1NFT*
- not always possible $\rightarrow$ decision procedure

Remarks:

- if local z-motion transductions are 1-way definable, then $\mathrm{squeeze}(T)$ can be defined
- iterate $\mathrm{squeeze}(T)$ $|Q|^2$ times (if possible), you get an equivalent 1-NFT

Extension to transducers

- same canvas (Rabin and Scott)
- removal of z-motion:
  ➜ translate a z-motion transducer into a *f1NFT*
- not always possible $\rightarrow$ decision procedure

Remarks:

- if local z-motion transductions are 1-way definable, then $\mathrm{squeeze}(T)$ can be defined
- iterate $\mathrm{squeeze}(T)$ $|Q|^2$ times (if possible), you get an equivalent 1-NFT

Results:

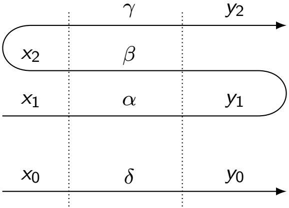
- decisition procedure to test whether a z-motion-transducer is 1-way definable
- the algorithm is complete

Decision procedure

Let T be a *f2NFT*.

1. repeat $|Q|^2$ times:
   - are all z-motion transductions of T *1NFT*-definable?
     - yes: $T \leftarrow \mathrm{squeeze}(T)$
     - no: STOP: the initial *2NFT* was not *1NFT*-definable!
2. remove backward transitions: you get an equivalent *1NFT*

Towards a characterization of 1-way definable z-motion-transductions



$$x_1 \cdot \alpha^n \cdot y_1 \cdot \beta^n \cdot x_2 \cdot \gamma^n \cdot y_2 = x_0 \cdot \delta^n \cdot y_0$$

Lemma (Fine and Wilf (56))

Let $u, v \in \Sigma^*$. If u^ω and v^ω have a sufficiently large common factor, then $u \in (w_1 w_2)^*$ and $v \in (w_2 w_1)^*$ for some $w_1, w_2 \in \Sigma^*$.

$\implies$ α,β,γ,δ have conjugate primitive roots (if $\neq \epsilon$).

$\rightarrow$ case analysis, depending on the emptiness of α, β, γ

Summary

Theorem

It is decidable whether a f2NFT is definable by a 1NFT.
If it is, one can construct an equivalent 1NFT.

- Complexity: non-elementary upper-bound, PSpace-hard.

Summary

Theorem

It is decidable whether a f2NFT is definable by a 1NFT.
If it is, one can construct an equivalent 1NFT.

- Complexity: non-elementary upper-bound, PSpace-hard.

Future Work

- Lower complexity (Shepherdson)
- What about *2NFT* (even non functional) ?
- Consider other structures: infinite strings, trees

Classes of Transductions

D="(input) deterministic"

f="functional"