Advanced Automata Theory 10 Transducers and Rational Relations

Frank Stephan

Department of Computer Science

Department of Mathematics

National University of Singapore

fstephan@comp.nus.edu.sg

Repetition: Automatic Structures

A structure $(\mathbf{A}, \mathbf{R_1}, \ldots, \mathbf{R_m}, \mathbf{f_1}, \ldots, \mathbf{f_n}, \mathbf{c_1}, \ldots, \mathbf{c_h})$ is automatic iff $\mathbf{A}$ is a regular set and each relation $\mathbf{R_k}$ is an automatic relation with domain $\mathbf{A}^{\ell_\mathbf{k}}$ and each function $\mathbf{f_k}$ is an automatic function mapping $\mathbf{A}^{\ell'_\mathbf{k}}$ to $\mathbf{A}$; the constants $\mathbf{c_1}, \ldots, \mathbf{c_h}$ are specific members of $\mathbf{A}$.

Examples

$(\mathbb{Z}, +, \mathbf{0})$ is a group with an automatic representation. Indeed, every fully automatic semigroup is by definition an automatic structure; automatic semigroups in sense of Epstein and coauthors are automatic structures of the form $(\mathbf{A}, \mathbf{f_1}, \ldots, \mathbf{f_n}, \varepsilon)$ where $\mathbf{A}$ is the set of representatives ("normal forms") of the semigroup and $\mathbf{f_1}, \ldots, \mathbf{f_n}$ are functions computing the semigroup multiplication with the fixed elements of a given finite set of generators of the semigroup.

$(\mathbb{Q}, +, \mathbf{0})$ has no automatic representation.

Repetition: Definability & Automaticity

Khoussainov and Nerode showed that whenever in an automatic structure a relation or function is first-order definable from other automatic relations or functions then it is automatic.

$(\mathbf{0^*}, \mathbf{Succ})$ with $\mathbf{Succ(w) = w0}$ is isomorphic to the structure $(\mathbb{N}, \mathbf{x \mapsto x + 1})$. The addition is not automatic in this structure, hence addition cannot be first-order defined from the successor-relation.

If $(\mathbf{A}, +, \mathbf{0})$ is isomorphic to $(\mathbb{N}, +, \mathbf{0})$ then $\mathbf{1}$ is uniquely determined in $\mathbf{A}$ by the axioms $\mathbf{1 \neq 0}$ and $\forall \mathbf{x}, \mathbf{y}\, [\mathbf{x + y = 1 \Rightarrow x = 0 \vee y = 0}]$. Hence one can define the ordering by $\mathbf{x < y \Leftrightarrow \exists z\, [x + z + 1 = y]}$.

Repetition: Rings and Semirings

A structure $(\mathbf{A}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$ is called a semiring with $\mathbf{1}$ iff it satisfies the following conditions:

1. $(\mathbf{A}, \oplus, \mathbf{0})$ is a commutative monoid;
2. $(\mathbf{A}, \otimes, \mathbf{1})$ is a monoid;
3. $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{A}\,[\mathbf{x} \otimes (\mathbf{y} \oplus \mathbf{z}) = (\mathbf{x} \otimes \mathbf{y}) \oplus (\mathbf{x} \otimes \mathbf{z})]$ and $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{A}\,[(\mathbf{x} \oplus \mathbf{y}) \otimes \mathbf{z} = (\mathbf{x} \otimes \mathbf{z}) \oplus (\mathbf{y} \otimes \mathbf{z})]$.

If, furthermore, $(\mathbf{A}, \oplus, \mathbf{0})$ is a group then $(\mathbf{A}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$ is called a ring with $\mathbf{1}$.

A semiring / ring is called commutative iff $\forall \mathbf{x}, \mathbf{y}\,[\mathbf{x} \otimes \mathbf{y} = \mathbf{y} \otimes \mathbf{x}]$.

Repetition: Infinite Automatic Ring

Assume that $(\mathbf{F}, +, *, \mathbf{0}, \mathbf{1})$ is a finite ring. Let $\mathbf{G}$ contain those elements $\mathbf{x_1 x_2 \ldots x_n}$ in $\mathbf{F}^*$ which either satisfy $\mathbf{n = 1}$ or $\mathbf{n > 1 \wedge x_{n-1} \neq x_n}$. Intuitively, $\mathbf{02112}$ stands for $\mathbf{021122222}\ldots$ where the last symbol repeats forever.

Now let $\mathbf{x_1 x_2 \ldots x_n + y_1 y_2 \ldots y_m = z_1 z_2 \ldots z_h}$ if for all $\mathbf{k > 0}$, $\mathbf{x}_{\min\{\mathbf{n},\mathbf{k}\}} + \mathbf{y}_{\min\{\mathbf{m},\mathbf{k}\}} = \mathbf{z}_{\min\{\mathbf{h},\mathbf{k}\}}$. Similarly for multiplication.

Now the member $\mathbf{0}$ of $\mathbf{F}$ is also the additive neutral element in $\mathbf{G}$ and $\mathbf{1}$ is also the multiplicative neutral element in $\mathbf{G}$.

The so generated $(\mathbf{G}, +, *, \mathbf{0}, \mathbf{1})$ is an example of an infinite automatic ring and represents the ring of the eventually constant functions $\mathbf{f} : \mathbb{N} \to \mathbf{F}$ with pointwise operations.

Repetition: Partial and Linear Orders

An ordering $\sqsubset$ on a set $\mathbf{A}$ is a relation satisfying the following two axioms:

1. $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{A}\, [\mathbf{x} \sqsubset \mathbf{y} \wedge \mathbf{y} \sqsubset \mathbf{z} \Rightarrow \mathbf{x} \sqsubset \mathbf{z}]$;
2. $\forall \mathbf{x}\, [\mathbf{x} \not\sqsubset \mathbf{x}]$.

Well-known automatic orderings are $<_{\mathbf{lex}}$, $<_{\mathbf{ll}}$, $<_{\mathbf{sh}}$ and $\prec$.

An ordering is called linear iff

3. $\forall \mathbf{x}, \mathbf{y} \in \mathbf{A}\, [\mathbf{x} \sqsubset \mathbf{y} \vee \mathbf{x} = \mathbf{y} \vee \mathbf{y} \sqsubset \mathbf{x}]$.

The orderings $<_{\mathbf{lex}}$ and $<_{\mathbf{ll}}$ are linear, the orderings $<_{\mathbf{sh}}$ and $\prec$ are not linear.

Repetition: Ordinals

Cantor designed a way to represent small ordinals as sums of descending chains of ω-powers: $\omega^{\mathbf{4}} + \omega^{\mathbf{2}} + \omega^{\mathbf{2}} + \omega$. Here $\omega^{\mathbf{k+1}}$ is the first ordinal which cannot be written as a finite sum of ordinals up to $\omega^{\mathbf{k}}$; ω is the first ordinal which cannot be written as $\mathbf{1} + \mathbf{1} + \ldots + \mathbf{1}$.

Write $\omega^{\mathbf{3}} \cdot \mathbf{2} + \omega \cdot \mathbf{3} + \mathbf{4}$ instead of $\omega^{\mathbf{3}} + \omega^{\mathbf{3}} + \omega + \omega + \omega + \mathbf{1} + \mathbf{1} + \mathbf{1} + \mathbf{1}$.

Example to add ordinals:
$(\omega^{\mathbf{8}} \cdot \mathbf{5} + \omega^{\mathbf{7}} \cdot \mathbf{2} + \omega^{\mathbf{4}}) + (\omega^{\mathbf{7}} + \omega^{\mathbf{6}} + \omega + \mathbf{1}) = \omega^{\mathbf{8}} \cdot \mathbf{5} + \omega^{\mathbf{7}} \cdot \mathbf{3} + \omega^{\mathbf{6}} + \omega + \mathbf{1}$.

Theorem of Delhomme: The ordinals below $\omega^{\mathbf{k}}$ with $\mathbf{k} \in \mathbb{N}$ have an automatic representation plus addition and comparison algorithm. This is impossible for larger sets of ordinals.

Rational Relations

Automatic Relation: Finite automaton reads all inputs involved at the same speed with $\#$ supplied for exhausted inputs.

Rational Relation: Nondeterministic finite automaton reads all inputs individually and can read them at different speed.

The first type of automata is called synchronous, the second type is called asynchronous.

There are many relations which are rational but not automatic.

Formal definition

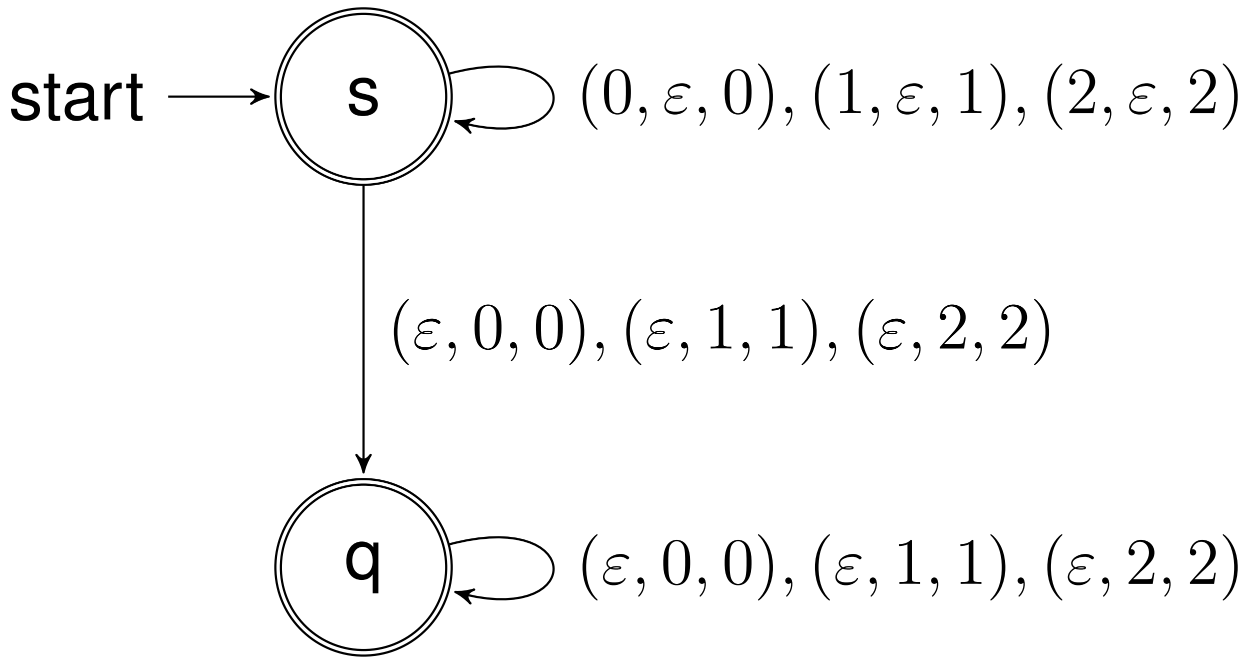
A rational relation $\mathbf{R} \subseteq (\Sigma^*)^{\mathbf{n}}$ is given by an non-deterministic finite state machine which can process $\mathbf{n}$ inputs in parallel and does not need to read them in the same speed. Transitions from one state $\mathbf{p}$ to a state $\mathbf{q}$ are labelled with an $\mathbf{n}$-tuple $(\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_n})$ of words $\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_n} \in \Sigma^*$ and the automaton can go along this transition iff for each input $\mathbf{k}$ the next $|\mathbf{w_k}|$ symbols in the input are exactly those in the string $\mathbf{w_k}$ (this condition is void if $\mathbf{w_k} = \varepsilon$) and in the case that the automaton goes on this transition, $|\mathbf{w_k}|$ symbols are read from the $\mathbf{k}$-th input word.

A tuple $(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$ is in $\mathbf{R}$ iff there is a run of the machine with transitions labelled by $(\mathbf{w_{1,1}}, \mathbf{w_{1,2}}, \ldots, \mathbf{w_{1,n}})$, $(\mathbf{w_{2,1}}, \mathbf{w_{2,2}}, \ldots, \mathbf{w_{2,n}})$, …, $(\mathbf{w_{m,1}}, \mathbf{w_{m,2}}, \ldots, \mathbf{w_{m,n}})$ ending up in an accepting state such that $\mathbf{x_1} = \mathbf{w_{1,1}} \mathbf{w_{2,1}} \ldots \mathbf{w_{m,1}}$, $\mathbf{x_2} = \mathbf{w_{2,1}} \mathbf{w_{2,2}} \ldots \mathbf{w_{m,2}}$, …, $\mathbf{x_n} = \mathbf{w_{1,n}} \mathbf{w_{2,n}} \ldots \mathbf{w_{m,n}}$.

Example 10.2: String Concatenation

Concatenation: $\mathbf{0100 \cdot 1122 = 01001122}$; $\mathbf{01 \cdot 210 = 01210}$; not an automatic relation.

The following automaton witnesses that it is a rational relation over alphabet $\mathbf{\Sigma = \{0, 1, 2\}}$.

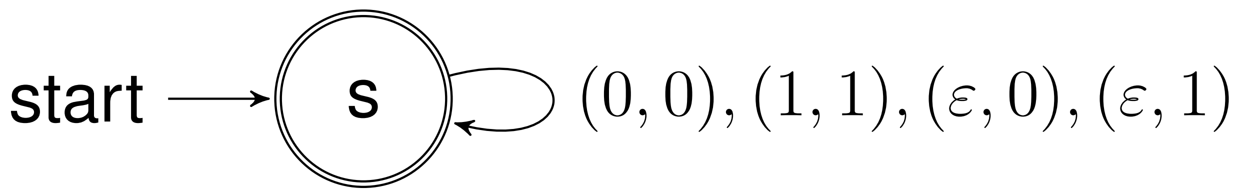


Sample Run: $\mathbf{(s01, s210, s01210) \Rightarrow (0s1, s210, 0s1210) \Rightarrow (01s, s210, 01s210) \Rightarrow (01q, 2q10, 012q10) \Rightarrow (01q, 21q0, 0121q0) \Rightarrow (01q, 210q, 01210q)}$.

Example 10.3: Subsequence

A string $\mathbf{x}$ is a subsequence of $\mathbf{y}$ iff it can be obtained by from $\mathbf{y}$ by deleting symbols at some positions. For example $\mathbf{12112}$ is a subsequence of $\mathbf{010200100102}$ and of $\mathbf{1211212}$ but not of $\mathbf{321123}$.

The following one-state automaton recognises this relation for the binary alphabet $\{\mathbf{0}, \mathbf{1}\}$.

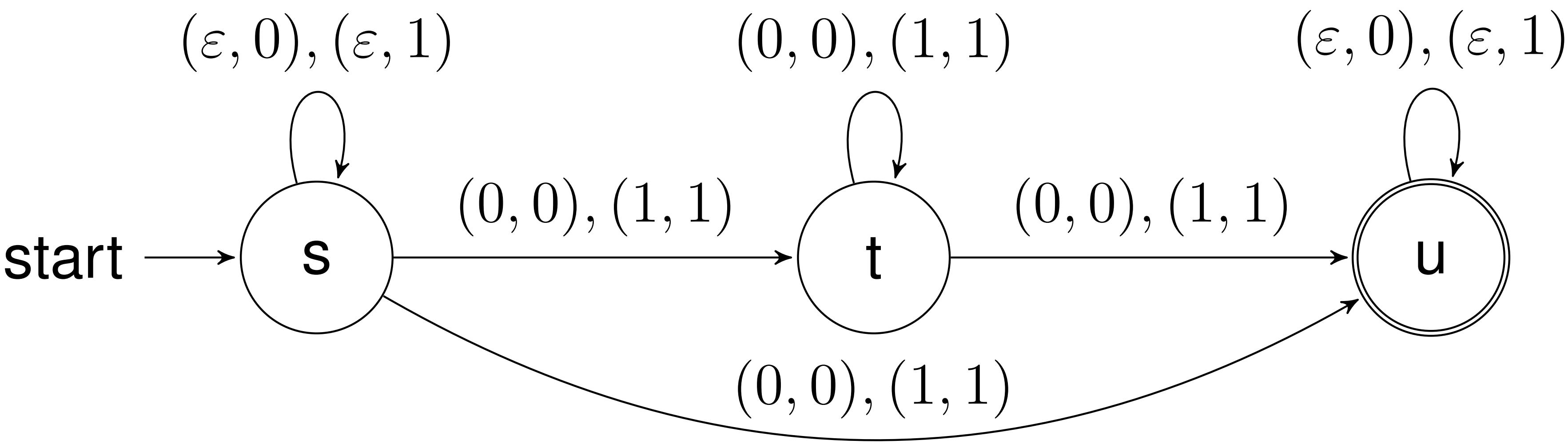


In general, there are one initial accepting state $\mathbf{s}$ with self-loops from $\mathbf{s}$ to $\mathbf{s}$ labelled with $(\varepsilon, \mathbf{a})$ and $(\mathbf{a}, \mathbf{a})$ for all $\mathbf{a} \in \Sigma$.

If $\mathbf{x} = \mathbf{0101}$ and $\mathbf{y} = \mathbf{00110011}$ then the automaton can accept this subsequence relation $(\mathbf{x}, \mathbf{y})$ using transitions labelled $(\mathbf{0}, \mathbf{0}), (\varepsilon, \mathbf{0}), (\mathbf{1}, \mathbf{1}), (\varepsilon, \mathbf{1}), (\mathbf{0}, \mathbf{0}), (\varepsilon, \mathbf{0}), (\mathbf{1}, \mathbf{1}), (\varepsilon, \mathbf{1})$.

Example 10.4: Substring

The following automaton recognises the relation of all $(\mathbf{x}, \mathbf{y})$ where $\mathbf{x}$ is a nonempty substring of $\mathbf{y}$, that is, $\mathbf{x} \neq \varepsilon$ and $\mathbf{y} = \mathbf{vxw}$ for some $\mathbf{v}, \mathbf{w} \in \{\mathbf{0}, \mathbf{1}\}^*$.

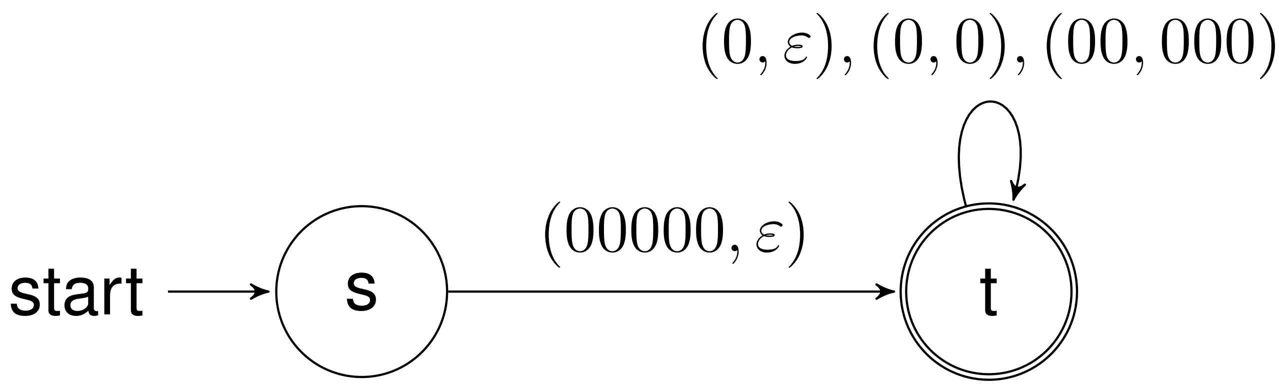


In $\mathbf{s}$: Parsing $(\varepsilon, \mathbf{v})$;
From $\mathbf{s}$ to $\mathbf{u}$: Parsing $(\mathbf{x}, \mathbf{x})$;
In $\mathbf{u}$: Parsing $(\varepsilon, \mathbf{w})$.

Quiz: Which labels must be added for $\mathbf{\Sigma} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}\}$?

Exercise 10.5: Rational Relations

Rational relations got their name, as one can use them in order to express relations between the various inputs words which are rational. For example, over alphabet $\{0\}$, the relation of all (x, y) with $|x| \geq \frac{2}{3}|y| + 5$ is recognised as follows:



Make automata which recognise the following relations:

(a) $\{(x, y) \in (0^*, 0^*) : 5 \cdot |x| = 8 \cdot |y|\}$;

(b) $\{(x, y, z) \in (0^*, 0^*, 0^*) : 2 \cdot |x| = |y| + |z|\}$;

(c) $\{(x, y, z) \in (0^*, 0^*, 0^*) : 3 \cdot |x| = |y| + |z| \vee |y| = |z|\}$.

Transducers

Transducers are automata recognising rational relations where the first parameter $\mathbf{x}$ is called input and the second parameter $\mathbf{y}$ is called output.

In particular one is interested in transducers computing functions: Two possible runs accepting $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}, \mathbf{z})$ must satisfy $\mathbf{y} = \mathbf{z}$. Given $\mathbf{x}$, there is a run accepting some pair $(\mathbf{x}, \mathbf{y})$ iff $\mathbf{x}$ is in the domain of the function.

Two main concepts:

Mealy machines:
Input/Output pairs on transition-labels.

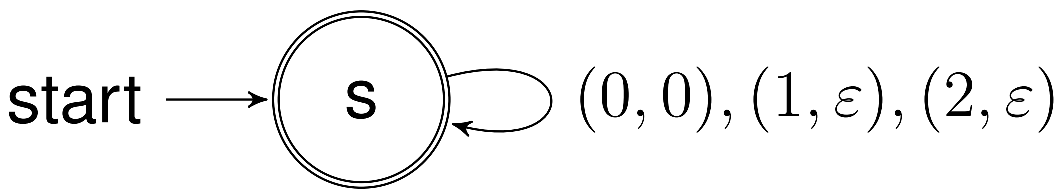
Moore machines:
Input on transition-labels; output in states.

Mealy machine

A Mealy machine computing a rational function $\mathbf{f}$ is a non-deterministic finite automaton such that each transition is attributed with a pair $(\mathbf{v}, \mathbf{w})$ of strings and whenever the machine follows a transition $(\mathbf{p}, (\mathbf{v}, \mathbf{w}), \mathbf{q})$ from state $\mathbf{p}$ to state $\mathbf{q}$ then one says that the Mealy machine processes the input part $\mathbf{v}$ and produces the output part $\mathbf{w}$.

Every automatic function is also a rational function and computed by a transducer, but not vice versa.

Mealy machine can compute π with $\pi(\mathbf{0}) = \mathbf{0}$, $\pi(\mathbf{1}) = \varepsilon$, $\pi(\mathbf{2}) = \varepsilon$, $\pi(\mathbf{v} \cdot \mathbf{w}) = \pi(\mathbf{v}) \cdot \pi(\mathbf{w})$:

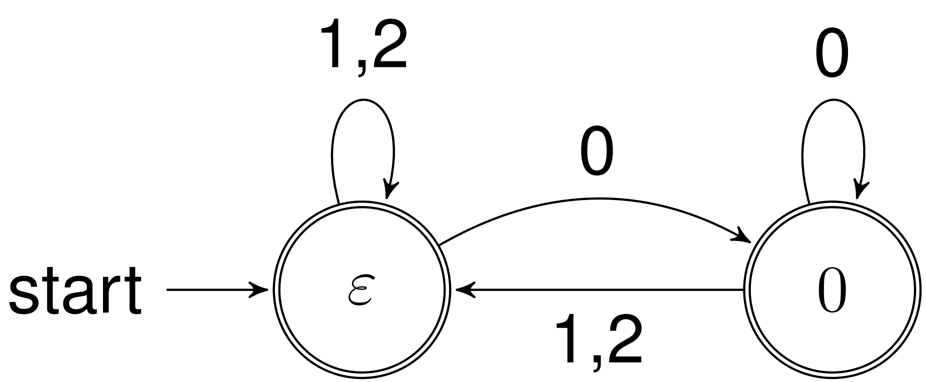


Moore machine

Moore machine

Non-deterministic finite automaton such that:
Possibly several starting states and final states;
Transitions (q, a, p) with input symbol $a \in \Sigma$;
States q labelled with output string $w_q \in \Sigma^*$;
Word $a_1 \ldots a_n$ translated into $w_{q_0} w_{q_1} \ldots w_{q_n}$ iff q_0 is a starting state and q_n is a final state and (q_m, a_{m+1}, q_{m+1}) is a valid transition for all $m < n$.

Moore machine erasing all $1, 2$ and preserving 0 computing function π with $\pi(012012) = 00$.



Example: Function f

Let $\mathbf{f(a_1a_2 \ldots a_n) = 012a_1a_1a_2a_2 \ldots a_na_n012}$, so $\mathbf{f(01) = 0120011012}$. Moore machine for $\mathbf{f}$:

state	starting	acc/rej	output	on 0	on 1	on 2
s	yes	rej	012	p, p'	q, q'	r, r'
p	no	rej	00	p, p'	q, q'	r, r'
q	no	rej	11	p, p'	q, q'	r, r'
r	no	rej	22	p, p'	q, q'	r, r'
s'	yes	acc	012012	–	–	–
p'	no	acc	00012	–	–	–
q'	no	acc	11012	–	–	–
r'	no	acc	22012	–	–	–

Quiz: Write Mealy machine for $\mathbf{f}$.

Example: Function g

Let $\mathbf{g(a_1 a_2 \ldots a_n) = (\max(\{a_1, a_2, \ldots, a_n\}))^n}$. So $\mathbf{g}(\varepsilon) = \varepsilon$, $\mathbf{g(000) = 000}$, $\mathbf{g(0110) = 1111}$ and $\mathbf{g(00212) = 22222}$.

Nodes $\{\mathbf{s, q_0, r_0, q_1, r_1, q_2, r_2}\}$;
Starting nodes: $\mathbf{s}$;
Accepting nodes: $\mathbf{s, r_0, r_1, r_2}$.
Output in nodes: $\mathbf{w_s} = \varepsilon$; $\mathbf{w_{q_0} = w_{r_0} = 0}$; $\mathbf{w_{q_1} = w_{r_1} = 1}$; $\mathbf{w_{q_2} = w_{r_2} = 2}$.
Transitions: $\mathbf{(s, 0, r_0)}$, $\mathbf{(s, 1, r_1)}$, $\mathbf{(s, 2, r_2)}$, $\mathbf{(s, 0, q_1)}$, $\mathbf{(s, 0, q_2)}$, $\mathbf{(s, 1, q_2)}$, $\mathbf{(r_0, 0, r_0)}$, $\mathbf{(q_1, 0, q_1)}$, $\mathbf{(q_1, 1, r_1)}$, $\mathbf{(r_1, 0, r_1)}$, $\mathbf{(r_1, 1, r_1)}$, $\mathbf{(q_2, 0, q_2)}$, $\mathbf{(q_2, 1, q_2)}$, $\mathbf{(q_2, 2, r_2)}$, $\mathbf{(r_2, 0, r_2)}$, $\mathbf{(r_2, 1, r_2)}$, $\mathbf{(r_2, 2, r_2)}$.

Comment: State $\mathbf{q_0}$ is unreachable and can be omitted; States $\mathbf{q_a}$ are for outputting $\mathbf{a}$ until $\mathbf{a}$ has been seen on the input, states $\mathbf{r_a}$ are for confirmed output $\mathbf{a}$. $\mathbf{s}$ is start state.

Quiz: Write a Mealy machine for $\mathbf{g}$.

Quiz

Determine the minimum number $\mathbf{m}$ such that every rational function can be computed by a non-deterministic Moore machine with at most $\mathbf{m}$ starting states and no ε-transition.

Note that $\mathbf{m}$ cannot be $\mathbf{1}$ as there is a function which maps ε to $\mathbf{0}$ and every non-empty word to $\mathbf{1}$. Give the Moore machine for this function.

Can the same be done to rule out $\mathbf{m = 2}$?

Exercises 10.10 and 10.11

A Moore machine / Mealy machine is deterministic, if it has exactly one starting state and each transition reads exactly one input symbol and for each pair (state,input symbol) there is at most one transition which applies.

Exercise 10.10. Make a deterministic Moore machine and a deterministic Mealy machine which flips input bits from $\mathbf{1}$ to $\mathbf{0}$ until it gets a $\mathbf{0}$ which is flipped to $\mathbf{1}$ and which from then onwards copies input to output.

Examples for input $\mapsto$ output are $\mathbf{0001} \mapsto \mathbf{1001}$, $\mathbf{110} \mapsto \mathbf{001}$, $\mathbf{1111} \mapsto \mathbf{0000}$, $\mathbf{0} \mapsto \mathbf{1}$ and $\mathbf{110011} \mapsto \mathbf{001011}$.

Exercise 10.11. Let the alphabet be $\{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$ and let $\mathbf{R} = \{(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}) : \mathbf{u}$ has $|\mathbf{x}|$ many $\mathbf{0}$s, $|\mathbf{y}|$ many $\mathbf{1}$s and $|\mathbf{z}|$ many $\mathbf{2}$s$\}$.

Is $\mathbf{R}$ a rational relation? Prove your result.

Theorem of Nivat

Theorem 10.12 [Nivat 1968]
Let $\mathbf{\Sigma_1, \Sigma_2, \ldots, \Sigma_m}$ be disjoint alphabets. Let $\pi_\mathbf{k}$ preserve the symbols from $\mathbf{\Sigma_k}$ and erase all other symbols.
Now a relation $\mathbf{R} \subseteq \mathbf{\Sigma_1^*} \times \mathbf{\Sigma_2^*} \times \ldots \times \mathbf{\Sigma_m^*}$ is rational iff there is a regular set $\mathbf{P}$ over a sufficiently large alphabet such that $(\mathbf{w_1, w_2, \ldots, w_n}) \in \mathbf{R} \Leftrightarrow \exists \mathbf{v} \in \mathbf{P}\,[\pi_\mathbf{1}(\mathbf{v}) = \mathbf{w_1} \wedge \pi_\mathbf{2}(\mathbf{v}) = \mathbf{w_2} \wedge \ldots \wedge \pi_\mathbf{m}(\mathbf{v}) = \mathbf{w_m}]$.

Example
Consider the subsequence relation where $\mathbf{\bar{x}_1 \bar{x}_2 \ldots \bar{x}_m}$ is a subsequence of $\mathbf{y_1 y_2 \ldots y_n}$ when deleting the overlines.
Now the regular set as requested by Nivat is, for ternary alphabet, $\mathbf{P} = \{\mathbf{0}, \mathbf{\bar{0}0}, \mathbf{1}, \mathbf{\bar{1}1}, \mathbf{2}, \mathbf{\bar{2}2}\}^*$.
Now $\overline{\mathbf{0010}}$ is subsequence of $\mathbf{00100}$ and $\mathbf{012012012012}$ but not of $\mathbf{220011}$. Here $\mathbf{\bar{0}0\bar{0}0\bar{1}1\bar{0}00}$ and $\mathbf{\bar{0}012\bar{0}0\bar{1}12\bar{0}012012}$ are in $\mathbf{P}$ and witness the two subsequence relations.

General Form of Theorem of Nivat

A relation $\mathbf{R}$ is rational iff there are functions (more precisely homomorphisms) $\pi_{\mathbf{1}}, \ldots, \pi_{\mathbf{n}}$ such that for each symbol $\mathbf{a}$ at most one $\pi_{\mathbf{k}}$ maps $\mathbf{a}$ to some nonzero word which consists of one symbol and a regular set $\mathbf{P}$ such that, for all $\mathbf{x_1}, \ldots, \mathbf{x_n}$, $\mathbf{R}(\mathbf{x_1}, \ldots, \mathbf{x_n})$ holds iff there is $\mathbf{y} \in \mathbf{P}$ with $\pi_{\mathbf{1}}(\mathbf{y}) = \mathbf{x_1}$, $\pi_{\mathbf{2}}(\mathbf{y}) = \mathbf{x_2}$, ..., $\pi_{\mathbf{n}}(\mathbf{y}) = \mathbf{x_n}$.

The subsequence relation of binary strings can be realised by $\pi_{\mathbf{1}}(\mathbf{0}) = \mathbf{0}, \pi_{\mathbf{1}}(\mathbf{1}) = \mathbf{1}, \pi_{\mathbf{2}}(\mathbf{2}) = \mathbf{0}, \pi_{\mathbf{2}}(\mathbf{3}) = \mathbf{1}$ and $\pi_{\mathbf{1}}, \pi_{\mathbf{2}}$ mapping all other symbols to ε and $\mathbf{x_1}$ is a subsequence of $\mathbf{x_2}$ iff there is a word $\mathbf{y} \in \{\mathbf{02}, \mathbf{13}, \mathbf{2}, \mathbf{3}\}^*$ with $\pi_{\mathbf{1}}(\mathbf{y}) = \mathbf{x_1}$ and $\pi_{\mathbf{2}}(\mathbf{y}) = \mathbf{x_2}$.

Example

The relation $\mathbf{R} = \{(\mathbf{0^n}, \mathbf{1^{n^2}}) : \mathbf{n} \geq \mathbf{1}\}$ is not rational.

To see this, one uses the Theorem of Nivat and considers a regular set $\mathbf{P}$ such that for each $\mathbf{n}$ there is a word $\mathbf{y} \in \mathbf{P}$ with $\mathbf{0^n} = \pi_\mathbf{1}(\mathbf{y})$ and $\mathbf{1^{n^2}} = \pi_\mathbf{2}(\mathbf{y})$.

As the set is regular, it satisfies the block pumping lemma with a constant $\mathbf{k}$. There is an $\mathbf{n}$ which is large enough so that $\mathbf{n^2} > (\mathbf{k} + \mathbf{1}) \cdot (\mathbf{n} + \mathbf{1})$. Thus there are at least $\mathbf{k} + \mathbf{1}$ many $\mathbf{1}$ without a $\mathbf{0}$ between them in any word $\mathbf{y} \in \mathbf{P}$ with $\mathbf{0^n} = \pi_\mathbf{1}(\mathbf{y})$ and $\mathbf{1^{n^2}} = \pi_\mathbf{2}(\mathbf{y})$. Thus one can cut the word into blocks such that all inner blocks contain each at least one $\mathbf{1}$ and no $\mathbf{0}$.

Now when one pumps up with the block pumping lemma, the number of $\mathbf{1}$ increases while the number of $\mathbf{0}$ remains the same. The pumping destroys $\mathbf{R}$ and $\mathbf{R}$ is not rational.

Rational Structures

A structure $(\mathbf{A}, \mathbf{R_1}, \mathbf{R_2}, \ldots, \mathbf{R_k}, \mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_h})$ is rational iff all the $\mathbf{A}$ is regular and $\mathbf{R_1}, \mathbf{R_2}, \ldots, \mathbf{R_k}$ are rational relations and $\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_h}$ are rational functions.

Furthermore, structures isomorphic to a rational structure might also be called rational.

Every automatic structure is by definition also a rational structure, but not vice versa.

The monoid $(\{\mathbf{0}, \mathbf{1}\}^*, \cdot, \varepsilon)$ with $\cdot$ being the concatenation is a rational structure but not an automatic structure, that is, not a fully automatic semigroup.

Theorem of Khoussainov and Nerode

The Theorem of Khoussainov and Nerode does not hold for rational structures.

- There are relations and functions which are first-order definable from rational relations without being a rational relation;
- There is no algorithm to decide whether a given first-order formula in a rational structure is true.

However, certain structures which are not automatic, are still rational.

Post Correspondence Problem

Let $\mathbf{I}$ be an index alphabet of at least two symbols and $\mathbf{f}, \mathbf{g} : \mathbf{I}^* \to \mathbf{I}^*$ rational functions which replace every symbol $\mathbf{e} \in \mathbf{I}$ by a word $\mathbf{f}(\mathbf{e})$ and $\mathbf{g}(\mathbf{e})$, respectively. Furthermore,

$$\mathbf{f}(\mathbf{e_1 e_2 \ldots e_n}) = \mathbf{f}(\mathbf{e_1}) \cdot \mathbf{f}(\mathbf{e_2}) \cdot \ldots \cdot \mathbf{f}(\mathbf{e_n})$$

and similarly for $\mathbf{g}$. Now the Post Correspondence Problem $(\mathbf{I}, \mathbf{f}, \mathbf{g})$ has a solution iff there is $\mathbf{u} \in \mathbf{I}^+$ with $\mathbf{f}(\mathbf{u}) = \mathbf{g}(\mathbf{u})$. It is undecidable whether any given instance of the Post Correspondence Problem has a solution.

Note that for fixed $\mathbf{f}, \mathbf{g}$, they are computed by transducers. Furthermore, equality is a rational relation. Thus the formula

$$\exists \mathbf{u} \in \mathbf{I}^* \, [\mathbf{u} \neq \varepsilon \wedge \mathbf{f}(\mathbf{u}) = \mathbf{g}(\mathbf{u})]$$

is first-order defined in the rational structure $(\mathbf{I}^*, \mathbf{f}, \mathbf{g}, =, \varepsilon)$.

Additional Explanation

Given $\mathbf{I} = \{0, 1, \ldots, k\}$, $\mathbf{f}(0), \mathbf{f}(1), \ldots, \mathbf{f}(k)$ and $\mathbf{g}(0), \mathbf{g}(1), \ldots, \mathbf{g}(k)$, one defines the transducer for $\mathbf{f}$ by a single accepting and starting state $\mathbf{s}$ with transitions labelled $(\mathbf{a}, \mathbf{f}(\mathbf{a}))$ for all $\mathbf{a} \in \mathbf{I}$. Similarly one verifies that $\mathbf{g}$ is rational.

Now if the first-order theory of rational structures would be decidable (in a uniform way as automatic structures), then there would be an algorithm which receives as input the finite automata of all functions, relations and sets involved and which receives a formula like

$$(\mathbf{PCP})\ \exists \mathbf{u} \in \mathbf{I}^* \left[\mathbf{u} \neq \varepsilon \wedge \mathbf{f}(\mathbf{u}) = \mathbf{g}(\mathbf{u})\right]$$

in a machine-readable form and then produces an output whether this formula is true or false in the structure. In this case, one would feed the automata for $\mathbf{f}, \mathbf{g}$ and some info on $\mathbf{I}$ and then a specialised algorithm would say whether the formula (PCP) is true or false.

Theory of Concatenation of Strings

A concrete rational structure with an undecidable first-order theory is $(\{\mathbf{0}, \mathbf{1}\}^*, \cdot, =, \prec, \varepsilon, \mathbf{0}, \mathbf{1})$ where $\cdot$ is the string concatenation of binary strings and $\prec$ is the prefix-relation on binary strings.

Lars Kristiansen and Juvenal Murwanashyaka provide on

https://arxiv.org/abs/1804.06367

an overview on how many quantifiers are needed to make undecidable formulas using existential and bounded quantifiers in the theory of binary strings with concatenation and string extension order. They in particular show that if the string extension order is there, then four existential quantifiers followed by some bounded quantifiers allow to make formulas whose truth in the theory cannot be decided. This is done by coding the Post Correspondence Problem in an adequate way.

Exercise 10.15: Random Graph

There is a rational representation of the random graph. Instead of coding (V, E) directly, one first codes a directed graph (V, F) with the following properties:

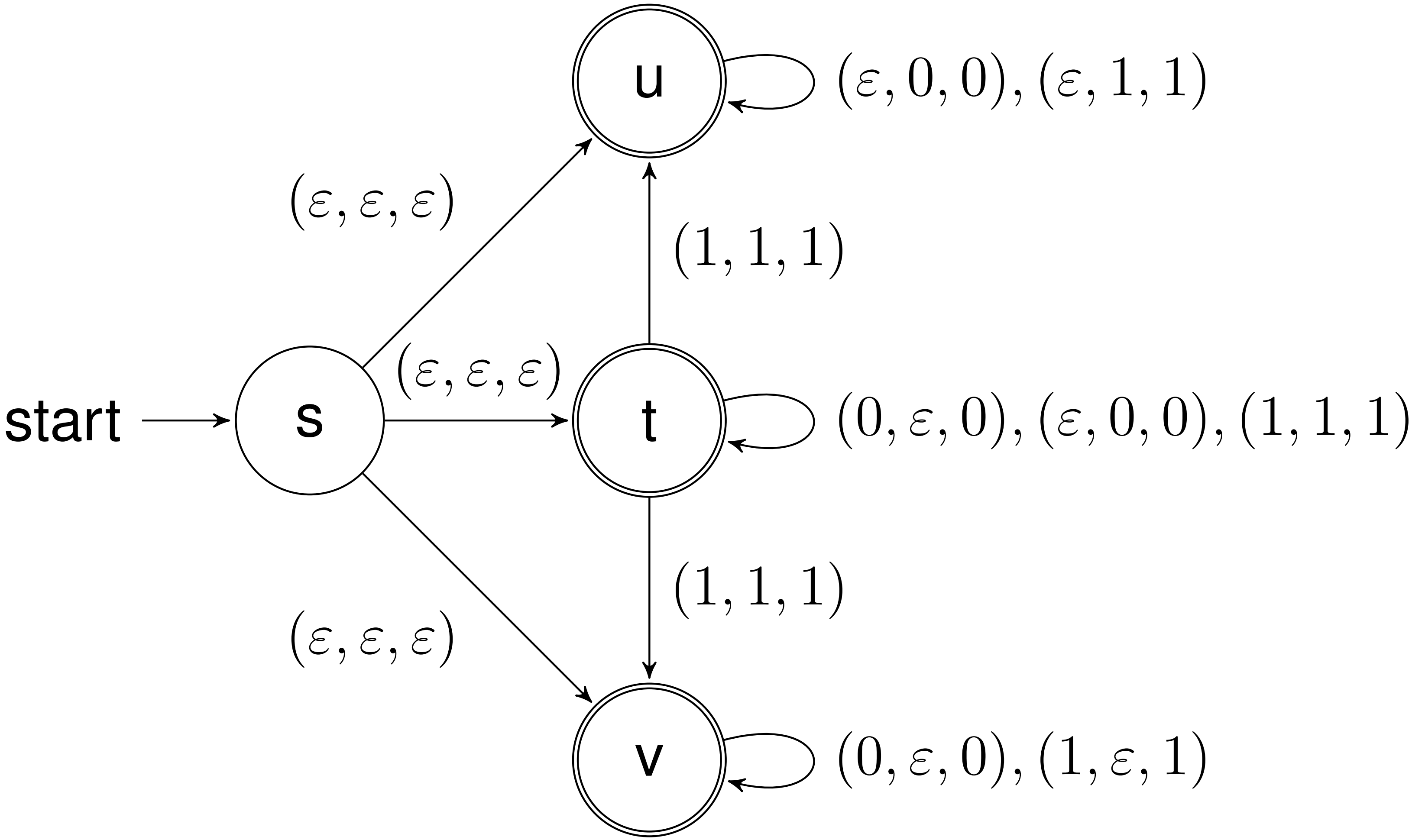
- For each $x, y \in V$, if $(x, y) \in F$ then $|x| < |y|/2$;
- For each finite $W \subseteq V$ there is a y with $\forall x\,[(x, y) \in F \Leftrightarrow x \in W]$.

This is done by letting $V = \{00, 01, 10, 11\}^+$ and defining that $(x, y) \in F$ iff there are n, m, k such that $y = a_0 b_0 a_1 b_1 \ldots a_n b_n$ and $a_m = a_k = 0$ and $a_h = 1$ for all h with $m < h < k$ and $x = b_m b_{m+1} \ldots b_{k-1}$. Give a transducer recognising F and show that this F satisfies the two properties above.

Now let $(x, y) \in E \Leftrightarrow (x, y) \in F \vee (y, x) \in F$. Show that (V, E) is the random graph.

Multiplication of Natural Numbers

The multiplicative monoid $(\mathbb{N} - \{0\}, *, 1)$ has a rational representation. Here one would represent $2^{n_1} \cdot 3^{n_2} \cdot \ldots \cdot p_k{}^{n_k}$ with $n_k > 0$ by $0^{n_1}10^{n_2}1\ldots0^{n_k}1$ and 1 by ε.



Exercises 10.17-10.21

Let $\mathbf{R}$ be a binary rational relation and let $\mathbf{L}$ be any language. Now define $\mathbf{R}(\mathbf{L}) = \{\mathbf{v} : \exists \mathbf{w} \in \mathbf{L}\, [\mathbf{R}(\mathbf{v}, \mathbf{w})]\}$. Similarly for a ternary rational relation $\mathbf{S}$, let $\mathbf{S}(\mathbf{L}, \mathbf{H}) = \{\mathbf{u} : \exists \mathbf{v} \in \mathbf{L}\, \exists \mathbf{w} \in \mathbf{H}\, [\mathbf{S}(\mathbf{u}, \mathbf{v}, \mathbf{w})]\}$.

Exercise 10.17: Show that if $\mathbf{L}$ and $\mathbf{H}$ are regular, so are $\mathbf{R}(\mathbf{L})$ and $\mathbf{S}(\mathbf{L}, \mathbf{H})$.

Exercise 10.18: Show that if $\mathbf{L}$ is context-free, so is $\mathbf{R}(\mathbf{L})$.

Exercise 10.19: If $\mathbf{L}, \mathbf{H}$ are context-free, is then also $\mathbf{S}(\mathbf{L}, \mathbf{H})$ context-free? Prove the answer.

Exercise 10.20: If $\mathbf{L}$ is context-sensitive, is so $\mathbf{R}(\mathbf{L})$?

Exercise 10.21: For which Boolean operations (union, intersection, set difference, symmetric difference) is there a rational relation $\mathbf{S}$ such that $\mathbf{S}(\mathbf{L}, \mathbf{H})$ is the corresponding combination of $\mathbf{L}$ and $\mathbf{H}$?

Exercise 10.22-10.26

Exercise 10.22: Is there a transducer Q which recognises the relation of all pairs $(0^n, 1^n 2^n)$ with $n \in \mathbb{N}$?

Exercise 10.23: Construct a transducer R which recognises a triple (u, v, w) iff v and w have a common subsequence of length at least $|u|$.

Exercise 10.24: Is there a transducer S which recognises a pair (v, w) iff v is the mirror image of w?

Exercise 10.25: Is there a transducer T which recognises a pair (v, w) iff v occurs in w two times as a subword?

Exercise 10.26: Is there a transducer U which recognises all pairs (v, w) such that in v, w occur the same symbols the same number of times?