Lecture IV

Transducers (1)

The 2-tape Turing machine model

This part which spans over the last two lectures studies the model of finite automata 'with output' which are usualy called 'transducers'. They can be seen as (finite) automata over a direct product of free monoids as well as (finite) automata over a free monoid with multiplicities in the (rational) subsets of another free monoid, or of a direct product of free monoids. These two models are also equivalent to 'one-way' Turing machines with two or more tapes. In this lecture, we consider the first model only, the second one is subject of the next lecture.

Contents

Automata 'with output' are a very natural, even a necessary, extension of automata that 'read' sequences of symboles. Since the dawn of automata theory (that is, the second half of the fifties), kinds of such automata with output were studied: *Moore machines* in which the sequences of states reached in the course of the reading of a word are observed, *Mealy machines* in which an output letter is associated with every transition. These two models are indeed equivalent up to some adjustment. We start with a model which is strictly more general.

1 Definitions

In the sequel, A and B are two alphabets. The set $A^* \times B^*$ of pairs (u,v) with u in A^* and v in B^*, equipped with the product:

$$(u,v)\,(u',v') = (u\,u', v\,v')$$

is a monoid, whose identity element is $(1_{A^*}, 1_{B^*})$, most often denoted by $(1,1)$. The *length* of an element of $A^* \times B^*$ is the sum of the lengths of its components: $|(u,v)| = |u| + |v|$. The monoid $A^* \times B^*$ is *graded* (Definition I.32). Similarly, $A_1^* \times A_2^* \times \cdots \times A_k^*$, the set of k-tuples of words equipped with the componentwise product is a graded monoid.

1.1 Transducers

Definition 1. A *transducer* is an automaton over $A^* \times B^*$ or, more generaly, over $A_1^* \times A_2^* \times \cdots \times A_k^*$, that is, an automaton whose transitions are labelled with k-tuples of words.

In (almost) all examples, $k = 2$. In the sequel, we also speak of 'pairs' rather than of 'k-tuple', unless stated otherwise.

A transducer is thus implicitely here a *Boolean automaton*[1] which can be denoted by $\mathcal{T} = \langle\, A^* \times B^*, Q, I, E, T \,\rangle$ where, as in the preceding lectures, Q is the state set, I and T are the sets of initial and final states respectively and where $E \subseteq Q \times (A^* \times B^*) \times Q$ is the set of transitions. Figure 1 shows four transducers.

We thus write $p \xrightarrow{(u,v)} q$ for a transition and

$$c = p_0 \xrightarrow{(u_1,v_1)} p_1 \xrightarrow{(u_2,v_2)} p_2 \cdots p_{n-1} \xrightarrow{(u_n,v_n)} p_n$$

for a computation of $\mathcal{T}$. The label of a computation is the product of the labels of its transitions and we write:

$$c = p_0 \xrightarrow{(u_1 u_2 \cdots u_n, v_1 v_2 \cdots v_n)} p_n \quad .$$

[1]We could have defined *weighted transducers* but their study is somewhat more complex and we need to know the theory of Boolean transducers first.

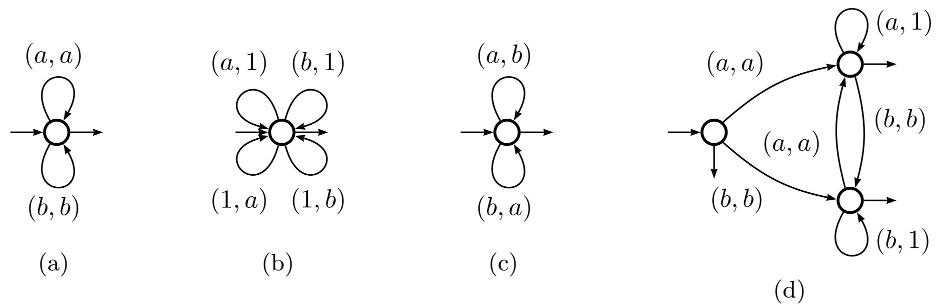


Figure 1: Four transducers

A computation is *successful* if its origin is an initial state and if its destination is a final state. A pair of words (u, v) in $A^* \times B^*$ is accepted by $\mathcal{T}$ if it is the label of a successful computation of $\mathcal{T}$. The *behaviour* of $\mathcal{T}$, denoted by $|\mathcal{T}|$, is the the set of pairs of words accepted by $\mathcal{T}$:

$$|\mathcal{T}| = \left\{ (u, v) \in A^* \times B^* \;\middle|\; \exists i \in I\,,\ \exists t \in T \quad i \xrightarrow[\mathcal{T}]{(u,v)} t \right\} \quad .$$

Examples 2. The transducer of Fig. 1 (a) accepts the set of pairs (u, u) where u is any word of $\{a, b\}^*$; the one of Fig. 1 (b) accepts the set of pairs (u, v) where u and v are any words of $\{a, b\}^*$; the one of Fig. 1 (c) accepts the set of pairs (u, v) where u is any word of $\{a, b\}^*$ and where v is obtained from u by replacing the a's by b's and the b's by a's; the one of Fig. 1 (d) accepts the set of pairs (u, v) where u is any word of $\{a, b\}^*$ and where v is obtained from u by replacing every block of a's by a unique a and every block of b's by a unique b.

The behaviour of a transducer is thus a subset of $A^* \times B^*$, that is, what we call a *relation between words*, or a *word relation*.

1.2 Word relations

Relations A relation θ from A^* to B^* is written (with a slight abuse) $\theta \colon A^* \to B^*$ and is *defined by its graph* $\widehat{\theta} \subseteq A^* \times B^*$. By definition, a relation from A^* to B^* associates with every word of A^* a subset of B^*:

$$\forall u \in A^* \quad \theta(u) = \left\{ v \in B^* \;\middle|\; (u, v) \in \widehat{\theta} \right\} \quad .$$

The underlying idea is that the first component of a pair (u, v) is an 'input' and that the second component is the 'output'. This point of view gives distinct roles to the two components of the pair but does not break the symmetry between them.

Inverse Indeed, A^* and B^* play symmetric roles by the way of the graph of θ and the *inverse relation* of θ,

$$\theta^{-1} \colon B^* \to A^* \quad ,$$

is defined as the relation from B^* to A^* which has the same graph as θ, modulo the canonical identification between $A^* \times B^*$ and $B^* \times A^*$:

$$\forall v \in B^* \qquad \theta^{-1}(v) = \left\{ u \in A^* \,\middle|\, (u,v) \in \widehat{\theta} \right\} \quad .$$

Additivity *By definition*, a relation is extended by *additivity*:

$$\forall L \subseteq A^* \quad \theta(L) = \bigcup_{u \in L} \theta(u)$$

and is thus viewed as an *application from* $\mathfrak{P}(A^*)$ *to* $\mathfrak{P}(B^*)$. Hence, the notion of relation implicitely carries with itself the property of additivity.

A contrario, complementation for instance, which associates with every subset of A^* a subset of A^*, *is not a relation* from A^* to itself.

Complement *By definition*, the *complement* of a relation $\theta\colon A^* \to B^*$ is the relation $\complement\theta\colon A^* \to B^*$ whose graph is the complement in $A^* \times B^*$ of the graph of θ:

$$\widehat{\complement\theta} = \complement\widehat{\theta} \qquad \text{that is,} \qquad \forall u \in A^* \qquad [\complement\theta](u) = \complement_{B^*}\theta(u) \quad .$$

Domain and image If $\theta\colon A^* \to B^*$ is a relation, the *domain* and the *image* of θ are the projections of $\widehat{\theta}$ onto A^* and B^* respectively:

$$\mathsf{Dom}\,\theta = \left\{ u \in A^* \,\middle|\, \exists v \in B^* \quad (u,v) \in \widehat{\theta} \right\} \qquad \text{and}$$
$$\mathsf{Im}\,\theta = \left\{ v \in B^* \,\middle|\, \exists u \in A^* \quad (u,v) \in \widehat{\theta} \right\} \quad .$$

Of course, $\mathsf{Dom}\,\theta^{-1} = \mathsf{Im}\,\theta$ and $\mathsf{Im}\,\theta^{-1} = \mathsf{Dom}\,\theta$. It also holds $u \notin \mathsf{Dom}\,\theta$ if and only if $\theta(u) = \emptyset$.

Generalisation to k-ary relations The relations of, or *predicats* on, $A_1^* \times A_2^* \times \cdots \times A_k^*$ — called *k-ary relations* — are defined by their graphs, which are subsets of $A_1^* \times A_2^* \times \cdots \times A_k^*$. What has been said above of *additivity*, inherent to the notion of relation, or of the *complement* of a relation, is naturally extended to k-ary relations. There are many ways[2] of 'currying' a relation of $A_1^* \times A_2^* \times \cdots \times A_k^*$ and the other notions: *domain*, *image*, *inverse* have a meaning only with respect to the way the 'input' and the 'output' components are chosen in the k-tuples elements de $A_1^* \times A_2^* \times \cdots \times A_k^*$. A natural generalisation is the one that could be denoted by $\theta\colon A_1^* \to A_2^* \times \cdots \times A_k^*$, where the input is a word of A_1^* and the output a $(k-1)$-tuple of words in $A_2^* \times \cdots \times A_k^*$.

[2] Properly speaking, 'currying' a function with several arguments consists in transforming it into a one-argument function which returns a function over the rest of the arguments.

1.3 Rational relations

The behaviour of a transducer is a subset of a direct product of free monoids; a transducer thus realises a relation, the one whose graph is the behaviour of this tranducer. For instance, the transducer of Fig. 1 (a) realises the identity function, the one of Fig. 1 (b) the universal relation. The Fundamental Theorem of Finite Automata yields a first characterisation of the relations realised by finite transducers. Let us first recall the definition of *rational subsets*, which holds in any monoid.

Definition 3. $\mathrm{Rat}\, A^* \times B^*$ is the smallest family of subsets of $A^* \times B^*$ which contains the finite subsets and which is closed under the operations of sum, product and star.

Let us recall also that a subset (of a monoid) is rational if and only if it is *denoted* by a rational expression.

Definition 4. A relation $\theta\colon A^* \to B^*$ is *rational* if so is its graph, that is, if $\widehat{\theta} \in \mathrm{Rat}\, A^* \times B^*$.

From the definition itself follows:

Property 5. *The inverse of a rational relation is a rational relation.*

The Fundamental Theorem of Finite Automata applied to transducers yields:

Theorem 6 (Elgot & Mezei 1965). *$\theta\colon A^* \to B^*$ is a rational relation if and only if $\widehat{\theta} = |\mathcal{T}|$ where $\mathcal{T}$ is a finite transducer over $A^* \times B^*$.*

If the label of every transition of a transducer $\mathcal{T}$ is mapped onto its first (resp. its second) component, one gets an automaton whose transitions are labelled by words — possibly *empty* — and which accepts the domain (resp. the image) of the relation realised by $\mathcal{T}$. This implies the following.

Corollary 7. $\theta\colon A^* \to B^*$ *rel. rat.* $\implies$ $\mathsf{Dom}\,\theta \in \mathrm{Rat}\, A^*$, $\mathsf{Im}\,\theta \in \mathrm{Rat}\, B^*$.

2 Working on the model and examples

The converse implication of Theorem 6 can, and must, be made more precise. In order to deal efficiently with transducers, it is convenient to have indeed a more constrained definition that does not diminish the power of the model, and also to be able to enrich it without making it more powerful.

2.1 Normalisation

The alphabet A *freely* 'generates' A^* since every word of A^* is the product of a *unique* sequence of letters of A. The set $(A \times \{1_{B^*}\}) \cup (\{1_{A^*}\} \times B)$ generates $A^* \times B^*$

since every pairs in $A^* \times B^*$ is the product of sequences of elements in $(A \times \{1_{B^*}\}) \cup (\{1_{A^*}\} \times B)$, but these sequences are not unique (in general):

$$(a\,b, b\,a\,b) = (a,1)\,(1,b)\,(1,a)\,(b,1)\,(1,b) = (1,b)\,(a,1)\,(b,1)\,(1,a)\,(1,b) \ \ .$$

One can also take $(A \times \{1_{B^*}\}) \cup (\{1_{A^*}\} \times B) \cup (A \times B)$ as generating set of $A^* \times B^*$; it allows to have shorter decomposition sequences:

$$(a\,b, b\,a\,b) = (a,b)\,(b,a)\,(1,b) = (1,b)\,(a,a)\,(b,b) \ \ .$$

The automata over A^* are defined with transitions labelled in A and it is known that the model is not more powerful, that is, does not accept more languages, if one allows labels in the whole A^*. For transducers, we follow a reverse process: they are defined with transitions whose labels are taken in the whole $A^* \times B^*$, and one shows that the model is not less powerful, that is, does not accept fewer relations, if the set of authorised labels is constrained.

Definition 8. (i) A transducer over $A^* \times B^*$ whose labels are in $(A \times \{1_{B^*}\}) \cup (\{1_{A^*}\} \times B)$ is called a *normalised transducer.*

(ii) A transducer over $A^* \times B^*$ whose labels are in $(A \times \{1_{B^*}\}) \cup (\{1_{A^*}\} \times B) \cup (A \times B)$ is called a *subnormalised transducer.*

The transducers (a), (c) and (d) of Figure 1 are subnormalised, the transducer (b) is normalised.

Proposition 9.
Every transducer is equivalent to a normalised (or subnormalised) transducer.

Proof. The process for transforming an arbitrary transducer into a normalised (or subnormalised) one is the same as in the case of automata over A^* labelled with words. It starts with the replacement of every transition whose label (u, v) is of length $\ell = |u|+|v|$ greater than 1 by ℓ transitions labelled in $(A \times \{1_{B^*}\}) \cup (\{1_{A^*}\} \times B)$ or by k transitions labelled in $(A \times \{1_{B^*}\}) \cup (\{1_{A^*}\} \times B) \cup (A \times B)$, with k contained between $\mathsf{max}(|u|, |v|)$ and ℓ.

In order to get a normalised, or subnormalised, transducer, it is necessary to eliminate the transitions which are labelled with $(1, 1)$, the identity element of $A^* \times B^*$, and whose presence is not ruled out by Definition 1. This elimination is the result of a classical algorithm which can be described in a slightly more general framework and which is worth to be explicitely given as it will be used later in an other construction.

Let M be a monoid. An *automaton over* M is a graph whose transitions are labelled with elements of M. Such an automaton is said to be *proper* if none of its transitions are labelled with the identity element of M.

Theorem 10.

Every finite automaton over M is equivalent to a proper *finite automaton.*

Proof. Let $\mathcal{A} = \langle M, Q, I, E, T \rangle$ be an automaton over M. We write: $E = F \cup S$ where S is the set of *spontaneous* transitions of $\mathcal{A}$, that is, the transitions labelled with 1_M. Without loss of generality, we assume that S is a *transitive* subgraph of $\mathcal{A}$: adding the transitions corresponding to the transitive closure of the set of spontaneous transitions in $\mathcal{A}$ may indeed change the computations of $\mathcal{A}$, but not their labels. A computation of $\mathcal{A}$ is then of the form:

$$c = p_0 \xrightarrow{m_1} p_1 \xrightarrow{m_2} p_2 \cdots p_{n-1} \xrightarrow{m_n} p_n \quad ,$$

and, thanks to the hypothesis on S, no two consecutive m_i are both equal to 1_M. Let $\mathcal{B} = \langle M, Q, J, G, T \rangle$ be the automaton defined by:

$$G = F \cup \{(p, m, r) \mid \exists q \in Q \quad (p, m, q) \in F \text{ and } (q, 1_M, r) \in S\} \quad \text{and}$$
$$J = I \cup \{j \mid \exists i \in I \quad (i, 1_M, j) \in S\}$$

which is then easily seen to be equivalent to $\mathcal{A}$. ■

This construction completes the proof of Proposition 9. ■

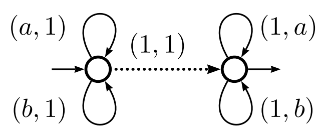
This result is also interesting in that it allows the use of spontaneous transitions for the construction of compact transducers as we see in the next series of examples.

Remark 11. The construction described in the proof of Theorem 10 can be called a *forward closure* as the new transitions are built with the spontaneous transitions that *follow* transitions labelled with elements different from the identity element. Another proper automaton equivalent to $\mathcal{A}$ can obviously be built by means of a dual *backward closure*.

2.2 Examples

Examples 12. (i) **Universal relation, direct product of rational sets**.

The universal relation, that is, the relation whose graph is the whole $A^* \times A^*$, is realised by the transducer of Figure 1 (b). It is also realised by the transducer below, in which every element of $A^* \times A^*$ is *the label of a unique computation* (and which demonstrates the benefit of spontaneous transitions).



If K is a language of A^*, accepted by $\mathcal{A}$, and L a language of B^*, accepted by $\mathcal{B}$, we transform $\mathcal{A}$ into a transducer $\mathcal{A}'$ by replacing the label 'a' of every transition by '$(a,1)$' and $\mathcal{B}$ into a transducer $\mathcal{B}'$ by replacing the label 'b' of every transition by '$(1,b)$' respectively, as shown below.

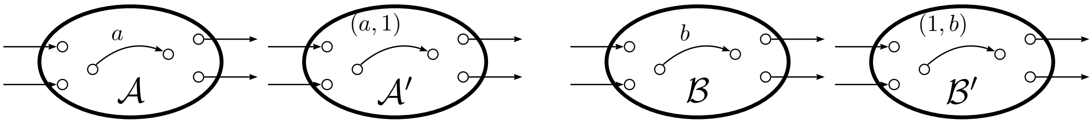


Figure 2: Transformation of automata into transducers

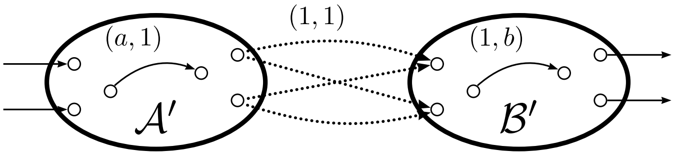
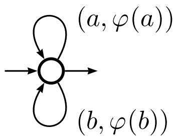


Figure 3: A transducer for $K \times L$

The transducer of Figure 3 realises the relation whose graph is $K \times L$.

(ii) **Identity, morphisms**. The identity, that is, the relation whose graph is $\{(w,w) \mid w \in A^*\}$, is realised by the transducer of Figure 1 (a). A *morphism* $\varphi\colon A^* \to B^*$ is realised by the transducer below.



(iii) **Intersection with a rational set**. If K is a language of A^*, the *intersection with* K is a relation from A^* into itself, denoted by ι_K, and defined by:

$$\forall w \in A^* \qquad \iota_K(w) = \begin{cases} w & \text{if } w \in K \\ \text{undefined (or } \emptyset) & \text{otherwise}. \end{cases}$$

If K is accepted by $\mathcal{A}$, the relation ι_K is realised by the transducer $\mathcal{A}''$ obtained from $\mathcal{A}$ by remplacing the label 'a' of every transition by '$(a,1)$', as shown below.

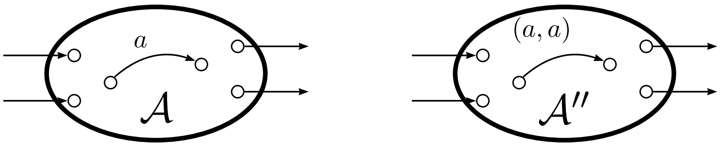


Figure 4: A transducer for ι_K

(iv) **Factors, subwords**. The relation from A^* into itself which associates with every word its *factors* is realised by the transducer shown at Figure 5 (a); the one which associates its *subwords* is realised by the transducer shown at Figure 5 (b).

(vi) **Operations on numbers written in base** p. When a base p is chosen, numbers (non-negative integers) are written[3] on the alphabet $A_p = \{0, 1, \ldots, p-1\}$ and operations on numbers are functions from A_p^*, or $(A_p^*)^2$, or $(A_p^*)^3$, *etc.* into A_p^*.

[3]When alphabets of digits are used, the empty word is written ε.

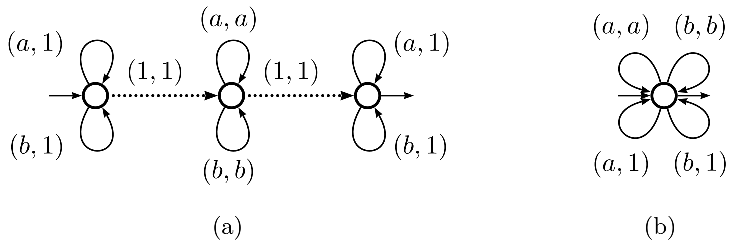


Figure 5: Factors and subwords

Some are realised by finite transducers. Figure 6 shows the example of the (integer) *division* by a fixed integer k, in the case where $p = 2$ and $k = 3$.

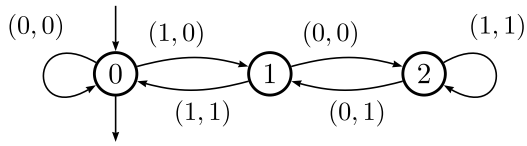


Figure 6: Integer division by 3 of numbers written in binary

2.3 Extension

2.3.1 k-ary transducers

As mentioned in Definition 1, a transducer may be an automaton over a direct product $A_1^* \times A_2^* \times \cdots \times A_k^*$ of k free monoids, not only an automaton over a direct product $A^* \times B^*$ of two free monoids. And as it has been mentioned as well, there are multiple ways of 'curryfying' a relation over $A_1^* \times A_2^* \times \cdots \times A_k^*$. From a theoretical point of view, it may be interesting to see such a relation as a function from A_1^* into the subsets of $A_2^* \times \cdots \times A_k^*$. From a practical point of view, it is more common to see the first $k-1$ components of a k-tuple as the 'input' and the k-th component as the result, that is, to view the relation as a map from $A_1^* \times A_2^* \times \cdots \times A_{k-1}^*$ into $\mathfrak{P}(A_k^*)$.

Example 13. Product in A^*. The relation $\pi\colon A^* \times A^* \to A^*$ which associates with every pair of words their product: $\pi(u, v) = uv$ for every u, v in A^*, is realised by the transducer below.

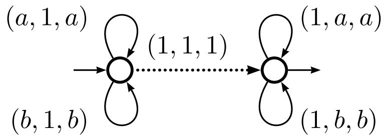


Figure 7: A 3-ary transducer for the product of words

The notions of *normalised* or *subnormalised* k-ary transducers are defined in an obvious manner and every k-ary transducer is equivalent to a normalised or subnormalised one (as is the transducer above if its spontaneous transition is eliminated).

It is useful to have this possible extension from 2 to k free monoids in mind but, as already said, we almost exclusively consider transducers over $A^* \times B^*$ in the sequel.

2.3.2 Left transducers, right transducers

A second variation on the model of transducers concerns the *direction of reading*. When we wrote that the *label of a computation is the product of the labels of the transitions* that make this computation (in an automaton or a transducer), it seemed understood that this product be from left to right. This corresponds to the *reading from left to right* in the machine model described in the next subsection. A reverse convention would have been as justified. There are even cases — as is Example 14 below — for which it is more natural.

This problem may be solved by means of *transposition*. The *transpose*, or *mirror image*, of a word $w = a_1 a_2 \cdots a_n$ is the word ${}^{\mathsf{t}}w = a_n \cdots a_2 a_1$; the transpose of a pair (u, v) is the pair $({}^{\mathsf{t}}u, {}^{\mathsf{t}}v)$. The transpose of an automaton, or of a transducer, $\mathcal{A} = \langle Q, I, E, T \rangle$, is the automaton, or the transducer, ${}^{\mathsf{t}}\mathcal{A} = \langle Q, T, {}^{\mathsf{t}}E, I \rangle$, avec

$${}^{\mathsf{t}}E = \{(p, {}^{\mathsf{t}}x, q) \mid (q, x, p) \in E\} \quad .$$

A word w is accepted by $\mathcal{A}$ in a *right-to-left reading* if and only if ${}^{\mathsf{t}}w$ is accepted by ${}^{\mathsf{t}}\mathcal{A}$ in a *left-to-right reading*. A word v belongs to the image of a word u in the relation realised by a transducer $\mathcal{A}$ in a *right-to-left reading* if and only if ${}^{\mathsf{t}}v$ belongs to the image of ${}^{\mathsf{t}}v$ par ${}^{\mathsf{t}}\mathcal{A}$ in a *left-to-right reading*.

In this way, it is seen that the inversion of the reading direction does not change the power of the model and does not bring anything new (as far as we have the transposition operator at hand). In some cases however, it may be simpler, more convenient or natural, to consider transducers *that read from right to left*, for instance when the transposed transducer is *input deterministic* as in Example 14 (what is called *right sequential transducer* in the last lecture.

Example 14. Addition in base 2. Let $A_2 = \{0, 1\}$. The map which associates with every pair (u, v) of words of $A_2 \times A_2$, that are the binary representations of the integers $\overline{u}$ and $\overline{v}$, the binary representation of $\overline{u} + \overline{v}$ is realised by the transducer of Figure 8 when it reads pairs *from right to left* (which is the usual way to perform addition indeed) and with the convention that the two words u and v are *justified on the right* and that the shorter one is padded with a sufficient number of '0' on the left to be of the same length as the longer one and, finally, that a last '0' is added on the left to both words in order to allow a last transition toward the final state (if necessary).

2.4 Transducers as machines

Modelling a finite Boolean automaton as a '1-way Turing machine' leads naturally to a generalisation of the model that features 'several tapes'. The machine consists

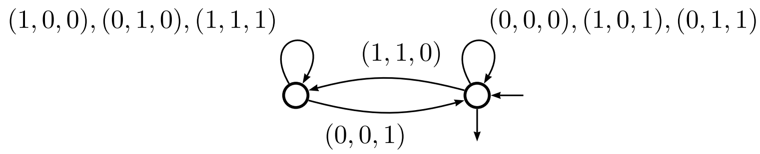


Figure 8: A 3-ary right transducer for the binary addition

of a finite state control unit and *several* tapes. The control unit is connected to every tape by a *reading head* (*cf.* Figure 9).

At every step of the computation, the control unit 'chooses', according to its state p, a tape on which it 'reads' and, depending on the symbol a read on the tape, jumps in state q and moves the reading head on that tape to the next cell on the right.[4] As reading heads are moved *always in the same direction*, this type of mahine is called *1-way* Turing machine.

At the beginning of a computation, a word is written on each of the k tapes, every reading head stays on the first cell of its tape and the control unit is in a distinguished state called *initial*. After a succession of steps, a computation ends if every reading head has reached on its tape the cell that contains the *end-of-tape* symbol. The computation is successful if at the end of the computation the control unit is in a state called *final*. A k-tuple of words is accepted by the machine if it can be read by a successful computation.

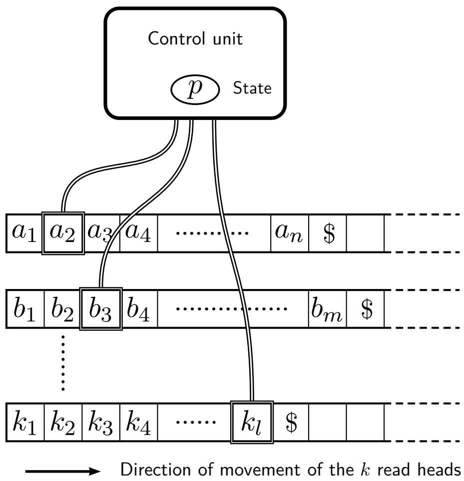


Figure 9: A k-tape 1-way Turing machine

Finite transducers over $A^* \times B^*$ are *strongly equivalent* to 1-way 2-tape Turing machines (1W2TTM) in the sense that for every transducer one can build such a machine which is not only equivalent (that is, accepts the same pair of words) but such that there is a bijection between their successful computations and *vice versa*.

[4]Other computation rules for such a device are possible. For instance, the choice of the read tape and of the destination state may depend not only on the state p but also on the symbols read on all tapes. All such definitions prove to be indeed equivalent.

The generalisation to transducers over $A_1^* \times A_2^* \times \cdots \times A_k^*$ and to 1-way k-tape Turing machines is tedious but conceals no difficulties.

3 Some facts

We review a series of negative results concerning rational sets of direct products of free monoids, hence rational relations. We end with an essential positive result that will be developped in Section 5: the closure by composition of rational relations. In the sequel, we call the finite transducers simply *transducers*.

3.1 Intersection, complement

In contrast with rational *languages*, rational *relations* are not closed under *intersection* and hence under *complement*.

Fact 15. $R, S \in \operatorname{Rat} A^* \times B^* \quad \not\Longrightarrow \quad R \cap S \in \operatorname{Rat} A^* \times B^*$.

Example 16. The behaviours of transducers[5] of Figure 10 are:

$$|\mathcal{V}_1| = \{(a^n b^m, c^n) \mid n, m \in \mathbb{N}\} \qquad \text{and} \qquad |\mathcal{W}_1| = \{(a^n b^m, c^m) \mid n, m \in \mathbb{N}\} \quad .$$

Hence $\qquad |\mathcal{V}_1| \cap |\mathcal{W}_1| = \{(a^n b^n, c^n) \mid n \in \mathbb{N}\} \;\notin\; \operatorname{Rat} \{a, b\}^* \times \{c\}^*$

since $\qquad \mathsf{Dom}\,(|\mathcal{V}_1| \cap |\mathcal{W}_1|) = \{a^n b^n \mid n \in \mathbb{N}\} \;\notin\; \operatorname{Rat} \{a, b\}^*$.

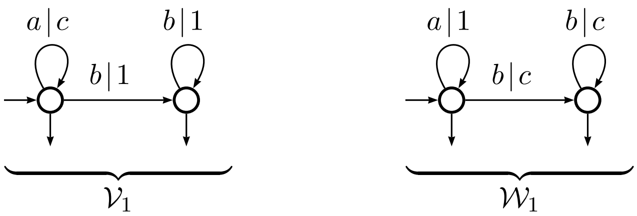


Figure 10: Transducers $\mathcal{V}_1$ and $\mathcal{W}_1$ over $\{a, b\}^* \times \{c\}^*$

Corollary 17. $\operatorname{Rat} A^* \times B^*$ *is not closed under complement.*

It holds nevertheless:

Proposition 18. *The complement of the identity is a rational relation.*

The proof reduces to the construction of the transducer of Figure 11 (and to the verification that its behaviour is indeed the complement of the identity).

[5] From this example on, we write $a \,|\, b$ instead of (a, b) for the labels of transitions, in order to lighten notation.

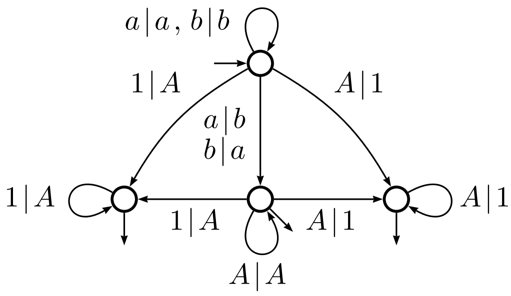


Figure 11: A transducer for the complement of the identity

3.2 Equivalence

A fundamental property of finite automata over a free monoid is that their *equivalence* is decidable, that is, there exists an algorithm which computes whether two such automata accept the same language. This property does not extend to rational relations.

Theorem 19 (Rabin & Scott 1959). *Let* $R, S \in \operatorname{Rat} A^* \times B^*$ *,* $\|A\|, \|B\| \geqslant 2$ *. It is undecidable whether* $R \cap S = \emptyset$ *or not.*

It follows:

Theorem 20 (Fischer & Rozenberg 1968).
The equivalence of finite transducers is undecidable.

These two negative results are established in the next section. Their statements leave open the status of the same questions in the cases where $\|A\| \geqslant 2, \|B\| = 1$ on one hand-side and $\|A\| = \|B\| = 1$ on the other. The first case exhibits an interesting separation between the two above statements.

Theorem 21 (Gibbons & Rytter 1986).
Let $R, S \in \operatorname{Rat} \{a,b\}^* \times \{c\}^*$ *. It is decidable whether* $R \cap S = \emptyset$ *or not.*

Theorem 22 (Ibarra 1978 – Lisovik 1979).
The equivalence of finite transducers over $\{a,b\}^* \times \{c\}^*$ *is undecidable.*

The second case pertains to a completely different theory. It is noticed that $\{a\}^* \times \{b\}^*$ is isomorphic to $\mathbb{N}^2$, the free commutative monoid with two generators. And it holds:

Theorem 23 (Ginsburg & Spanier 1966).
$\operatorname{Rat} \mathbb{N}^k$ *is an effective Boolean algebra, for every integer* k*.*

The proof of these last three results exceeds the program of these lectures (*cf.* EAT). We end this negative list with a positive result, not so much for cheering up, but because we need it in the next section for the proof of undecidability results.

3.3 Composition

The composition of functions directly extends to the one of (2-ary) relations.

Definition 24. Let $\theta\colon A^* \to B^*$ et $\sigma\colon B^* \to C^*$ two relations. The *composition* of θ and σ is the relation

$$\sigma\circ\theta\colon A^* \to C^* \qquad \text{defined by} \qquad \forall u \in A^* \qquad [\sigma\circ\theta](u) = \sigma(\theta(u)) \enspace .$$

The composition of relations can be defined (or expressed) by means of their graph:

$$\widehat{\sigma\circ\theta} = \left\{(u,w) \in A^*\times C^* \,\middle|\, \exists v \in B^* \quad (u,v) \in \widehat{\theta} \text{ and } (v,w) \in \widehat{\sigma}\right\} \enspace .$$

Theorem 25 (Elgot & Mezei 1965)**.**
The composition of two rational relations is a rational relation.

We come back to, and establish, this fundamental result in Section 5.

4 Undecidability results

The undecidable property par excellence is the 'halting problem for a Turing machine'. But one can take as a basis any other propery already proved to be undecidable. The one we shall use in the sequel, because it is simpler to state, and easier to deal with in connection with automata, is known as the 'Post Correspondence Problem'.

The Post Correspondence Problem (PCP)

Let B be an alphabet with at least two letters. Given an integer k and two sets of k words of B^*: $\{u_1, u_2, \ldots, u_k\}$ et $\{v_1, v_2, \ldots, v_k\}$, does there exist a sequence of indices $i_1, \ldots, i_p$ in $[k]$ such that

$$u_{i_1} u_{i_2} \cdots u_{i_p} = v_{i_1} v_{i_2} \cdots v_{i_p}\,?$$

Theorem 26 (Post 1946)**.** *(PCP) is recursively undecidable.*

This statement holds for the problem in full generality. If one looks for its status according to the number k that allows to formulate an instance, the situation is more complex. Let (PCP_k) be the above problem in which the integer k is fixed. It is known that (PCP_2) is decidable and, since recently, that (PCP_k) is undecidable for $k \geqslant 5$. The status of (PCP_k) is still open for k equal to 3 or 4.

Translation in the vocabulary of Language and Automata Theory

The reason for our choice is that (PCP) can be easily expressed in terms of *morphisms between free monoids*.

If $U = \{u_1, u_2, \ldots, u_k\}$ is given, we write: $A_k = \{1, 2, \ldots, k\}$, and

$$\tau_U \colon A_k^* \to B^* \qquad \text{for the morphism defined by} \quad \tau_U(i) = u_i \quad \text{for every } i \text{ in } [k] \ .$$

Similarly, if $V = \{v_1, v_2, \ldots, v_k\}$, we write: $\tau_V \colon A_k^* \to B^*$ the morphism defined by $\tau_V(i) = v_i$ for every i in $[k]$. A 'sequence of indices' is a word of A_k^* and (PCP) is rephrased into:

does there exist a word w in A_k^* such that $\tau_U(w) = \tau_V(w)$?

Theorem 26 then becomes:

Theorem 27. *Let θ and $\mu \colon A^* \to B^*$ be two morphisms. It is undecidable whether there exists w in A^* such that $\theta(w) = \mu(w)$ or not.*

Proof of Theorem 19. Let U and V be two sets of k words of B^* which produce an undecidable instance of (PCP) and $\tau_U \colon A_k^* \to B^*$ and $\tau_V \colon A_k^* \to B^*$ the corresponding morphisms.

To state that it is undecidable whether there exists w in A_k^* such that $\tau_U(w) = \tau_V(w)$ is equivalent as to state that it is undecidable whether

$$\widehat{\tau_U} \bigcap \widehat{\tau_V} = \emptyset \ ,$$

and τ_U and τ_U are rational relations (Example 16 (ii)). It remains to show that Theorem 19 holds for an alphabet $A = \{a, b\}$ with two letters only.

Let $\kappa \colon A_k^* \to A^*$ an *injective morphism* (defined, for instance, by $\kappa(i) = a^i b$). By Theorem 25, $\tau_U \circ \kappa^{-1}$ and $\tau_V \circ \kappa^{-1}$ are rational relations and, since κ is injective, it holds:

$$\widehat{\tau_U \circ \kappa^{-1}} \bigcap \widehat{\tau_V \circ \kappa^{-1}} = \emptyset \iff \widehat{\tau_U} \bigcap \widehat{\tau_V} = \emptyset \ . \qquad \blacksquare$$

Theorem 20 is a direct consequence of the following, more precise, statement.

Theorem 28. *Let $R \in \mathrm{Rat}\, A^* \times B^*$, $\|A\|, \|B\| \geq 2$. It is undecidable whether $R = A^* \times B^*$ or not.*

We first prove:

Lemma 29. *Let $\theta \colon A^* \to B^*$ be a* functional *rational relation. Then $\complement\theta \colon A^* \to B^*$ is a rational relation.*

Proof. Let χ be the complement of the identity on B^*, a rational relation by Proposition 18. We have:

$$\widehat{\complement\theta} = [(A^* \setminus \mathsf{Dom}\, \theta) \times B^*] \ \cup \ \widehat{\chi \circ \theta} \ .$$

The first term of the union is rational (Example 16 (i)) and so is the second one by Theorem 25. $\blacksquare$

Proof of Theorem 28. With the notation of the proof of Theorem 19, $\tau_U \circ \kappa^{-1}$ and $\tau_V \circ \kappa^{-1}$ are functional rational relations and it holds:

$$\complement\left(\widehat{\tau_U \circ \kappa^{-1}}\right) \bigcup \complement\left(\widehat{\tau_V \circ \kappa^{-1}}\right) = A^* \times B^* \iff \widehat{\tau_U \circ \kappa^{-1}} \bigcap \widehat{\tau_V \circ \kappa^{-1}} = \emptyset \ . \quad \blacksquare$$

5 Composition and evaluation

The closure by composition of rational relations (Theorem 25) is a fundamental property, as is the consequence we deduce from it: the Evaluation Theorem (Theorem 34).[6] Together, they make of rational relations a powerful tool for the classification of formal languages. But above all, they give its consistency to the model of transducers.

5.1 The Composition Theorem

Theorem 25 (Elgot & Mezei 1965)**.**

$$\theta\colon A^* \to B^*\,, \quad \sigma\colon B^* \to C^* \quad \textit{rat. rel.} \quad \Longrightarrow \quad \sigma\circ\theta\colon A^* \to C^* \quad \textit{rat. rel.}$$

Proof. Let $\mathcal{T} = \langle\, A^*{\times}B^*, Q\, I, E, T\,\rangle$ and $\mathcal{S} = \langle\, B^*{\times}C^*, R, J, F, U\,\rangle$ be two *subnormalised* transducers which realise θ and σ respectively. We define a *composition product* of transducers $\mathcal{U} = \mathcal{T}\circ\mathcal{S}$ by:

$$\mathcal{U} = \langle\, A^*{\times}C^*, Q{\times}R, I{\times}J, G, T{\times}U\,\rangle \qquad\qquad \text{with}$$

$$\begin{aligned} G = &\left\{(p,r) \xrightarrow{x|y} (q,s) \;\middle|\; \exists b\in B\,,\ \exists p \xrightarrow{x|b} q\in E\,,\ \exists r \xrightarrow{b|y} s\in F \quad x\in A\cup 1\,,\ y\in C\cup 1\right\}\\ &\bigcup\ \left\{(p,r) \xrightarrow{a|1} (q,r) \;\middle|\; \exists p \xrightarrow{a|1} q\in E \qquad \forall r\in R\right\}\\ &\qquad\bigcup\ \left\{(p,r) \xrightarrow{1|c} (p,s) \;\middle|\; \exists r \xrightarrow{1|c} s\in F \qquad \forall p\in Q\right\}\ . \end{aligned}$$

By induction on the length of the computation, it is verified that:

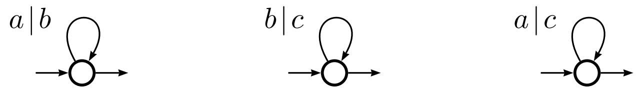
$$(p,r) \xrightarrow[\mathcal{U}]{u|w} (q,s) \quad \text{if and only if} \qquad \exists v \qquad p \xrightarrow[\mathcal{T}]{u|v} q \quad \text{and} \quad r \xrightarrow[\mathcal{S}]{v|w} s\ ,$$

which establish $|\mathcal{U}| = \widehat{\sigma\circ\theta}$. ∎

This composition product may yield a transducer $\mathcal{U}$ with some transitions that are labelled by $1|1$ (in the first group, when x *and* y are both equal to 1). These spontaneous transitions are eliminated (by 'backward' or 'forward' closure, for instance) in order to obtain a subnormalised transducer. In the sequel, it will be this subnormalised transducer which will be denoted by $\mathcal{U} = \mathcal{T}\circ\mathcal{S}$.

Example 30 (trivial)**.**

$$\mathcal{U}_1 = \mathcal{T}_1 \circ \mathcal{S}_1$$

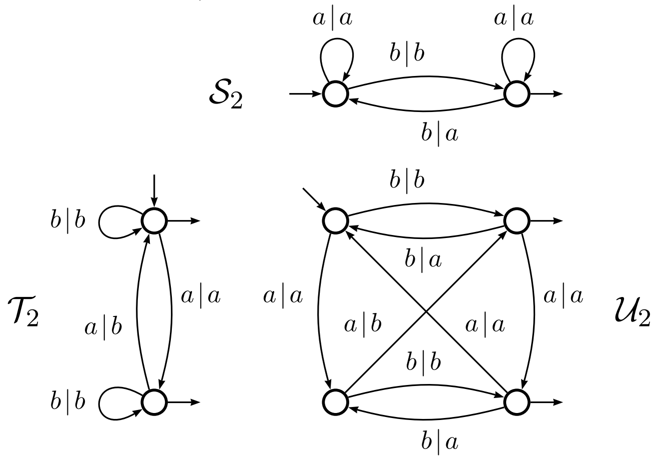


$\mathcal{T}_1$ $\mathcal{S}_1$ $\mathcal{U}_1$

[6] In the next lecture, we proceed in the reverse way: we first establish the Evaluation Theorem from wich we deduce the Composition Theorem.

Example 31 (less trivial but still simple)**.**

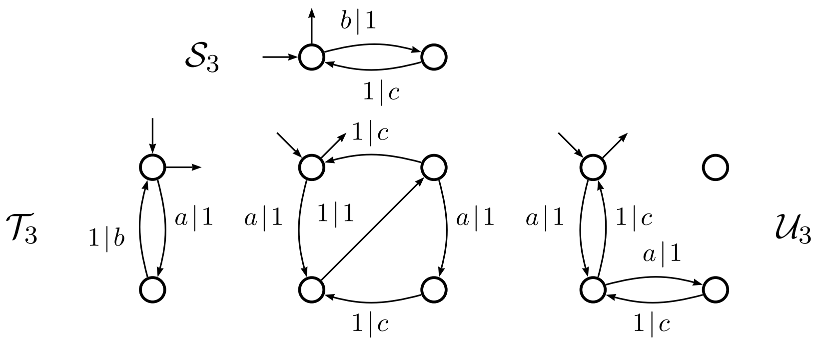
$\mathcal{U}_2 = \mathcal{T}_2 \circ \mathcal{S}_2$



These two examples show 'letter-to-letter transducers'. More general examples will be considered in the exercises.

Remark 32. If we consider the *normalised* transducers $\mathcal{T}_3$ and $\mathcal{S}_3$ that are equivalent to $\mathcal{T}_1$ and $\mathcal{S}_1$ respectively, a spontaneous transition appears in the course of the construction of the composition product $\mathcal{U}_3 = \mathcal{T}_3 \circ \mathcal{S}_3$. It is also important to note that in this case also, the *multiplicity* of computations is *not preserved.* A more elaborate construction (that is a more sophisticated composition product) allows to overcome this problem.

$\mathcal{U}_3 = \mathcal{T}_3 \circ \mathcal{S}_3$



Remark 33. It is possible to define (finite) transducers on direct products of *non free* monoids and hence *rational relations* between non necessarily free monoids: relations from M to N whose graph is in $\mathrm{Rat}\, M \times N$.

Two such relations $\theta \colon M \to N$ and $\sigma \colon N \to P$ can be then be composed. The construction of the proof of Theorem 25 still holds true — and the composition is a rational relation — as long as N is a free monoid B^*, but the composition may well be a non rational relation *if N is not a free monoid.*

For instance, let $\theta \colon \{a\}^* \to a^* \times b^*$ be the morphism defined by $\theta(a) = (a, b)$ and $\sigma \colon a^* \times b^* \to \{a, b\}^*$ the relation whose graph is $\widehat{\sigma} = \big((a,1), a\big)^* \big((1,b), b\big)^*$. Then $\sigma((a^n, b^m)) = a^n b^m$ holds and $\mathsf{Im}\,(\sigma \circ \theta) = \{a^n b^n \mid n \in \mathbb{N}\}$. It follows that $\sigma \circ \theta \colon \{a\}^* \to \{a, b\}^*$ is not a rational relation.

5.2 Two consequences

From Theorem 25 we deduce two important results: the Evaluation Theorem, and a 'restriction theorem'.

Theorem 34.

The image of a rational language by a rational relation is a rational language.

Proof. We want to prove: $\theta\colon A^* \to B^*$ rational relation and K in $\mathrm{Rat}\, A^*$ imply that $\theta(K)$ is in $\mathrm{Rat}\, B^*$. The following sequence of equalities holds.

$$\begin{aligned}\theta(K) &= \bigcup_{v\in K} \theta(v) = \bigcup_{v\in K} \left\{w \in B^* \,\middle|\, (v,w) \in \widehat{\theta}\right\}\\ &= \left\{w \in B^* \,\middle|\, \exists v \in K \quad (v,w) \in \widehat{\theta}\right\}\\ &= \left\{w \in B^* \,\middle|\, \exists u \in A^*\,,\ \exists v \in K \quad (u,v) \in \widehat{\iota} \text{ and } (v,w) \in \widehat{\theta}\right\}\\ &= \mathsf{Im}\ (\theta \circ \iota)\end{aligned}$$

∎

This result can be seen as a particular case of the following statement which contrasts with Fact 15.

Theorem 35. *Let $\theta\colon A^* \to B^*$ be a rational relation, K a rational language of A^* and L a rational language of B^*. Then $\widehat{\theta}\cap(K\times L)$ is the graph of a rational relation.*

Proof. It is easily verified that $\widehat{\theta} \cap (K\times L) = \widehat{\iota_L \circ \theta \circ \iota_K}$. ∎

Remark 36. Theorem 35 can also be seen as a particular instance of a general result on rational and *recognisable* subsets of non free monoids: the intersection of a rational and of a recognisable subsets of an arbitrary monoid is a rational subset and $K\times L$ is a recognisable subset of $A^*\times B^*$.

6 Exercises

1. **Orders.** The alphabet A is *totally ordered* and this order is denoted by $\leqslant$.

The *lexicographic order*, denoted by $\preccurlyeq$, extends the order on A to an order on A^* and is defined as follows. Let v and w be two words in A^* and u their *longest common prefix.* Then, $v \preccurlyeq w$ if $v = u$ or, if $v = u\,a\,s$, $w = u\,b\,t$ with a and b in A, then $a < b$.

(a) Give a finite transducer over $A^*\times A^*$ which realises $\preccurlyeq$, that is, which asssociates with every word u of A^* the set of words which are equal to or greater than u.

The *radix order* (also called the *genealogical order* or the *short-lex order*), denoted by $\sqsubseteq$, is defined as follows: $v \sqsubseteq w$ if $|v| < |w|$ or $|v| = |w|$ and $v \preccurlyeq w$.

Beware. new notation, but never used!

(b) Give a finite transducer over $A^*\times A^*$ which realises $\sqsubseteq$, For every language L of A^*, we denote by $\mathsf{minlg}\,(L)$ (resp. $\mathsf{Maxlg}\,(L)$) the set of words of L which have no smaller (resp. no greater) words in L of *the same length* in the lexicographic order.

(c) Show that if L is a rational language, so are $\mathsf{minlg}\,(L)$ and $\mathsf{Maxlg}\,(L)$.

2. **Number representation.**

Let $A_2 = \{0,1\}$ and $A_3 = \{0,1,2\}$ be two alphabets of digits.

The alphabet A_3 can be first considered as a non-canonical alphabet for the representation of integers in base 2: $\overline{1\,2} = 4$, $\overline{2\,0\,1} = 9$, *etc.*

Let $\nu_2 \colon A_3^* \to A_2^*$ be the *normalisation* in base 2, that is, the relation which associates with a word of A_3^* the word of A_2^* which represents the same integer in base 2.

(a) Give a transducer which realises ν_2. Comment.

Let $\varphi\colon A_2^* \to A_3^*$ be the function which maps the binary representation of every integer onto its representation in base 3, *e.g.* $\varphi(1\,0\,0\,0) = 2\,2$.

(b) Show that φ is not a rational relation.

3. **Operation on numbers.**

(a) Give a transducer which realises the multiplication by 9 on the integers written in binary representation, that is, the relation $\tau\colon A_2^* \to A_2^*$ such that $\overline{\tau(w)} = 9 \cdot \overline{w}$.

(b) Let $\mu\colon A_2^* \times A_2^* \to A_2^*$ be the relation which realises the multiplication, that is, such that $\mu(u,v) = w$ where $\overline{w} = \overline{u} \cdot \overline{v}$.
Show that μ is not a rational relation.

4. **Map equivalence of a morphism.**

Let $\varphi_1 \colon \{a,b,c\}^* \to \{x,y\}^*$ be the morphism defined by:

$$\varphi_1(a) = x\,, \quad \varphi_1(b) = y\,x\,, \quad \varphi_1(c) = x\,y\,.$$

(a) Give a subnormalised transducer which realises φ_1.

(b) Give a subnormalised transducer which realises ${\varphi_1}^{-1}$.

(c) Compute a subnormalised transducer which realises ${\varphi_1}^{-1} \circ \varphi_1$.

5. **Iteration lemma.** Let $\theta\colon A^* \to B^*$ be a rational relation.

(a) Show that there exists an integer N such that for every pair (u,v) in $\widehat{\theta}$ whose length[7] is greater than N, there exists a factorisation:

$$(u,v) = (s,t)\,(x,y)\,(w,z)$$

such that: (i) $1 \leqslant |x| + |y| \leqslant N$ and (ii) $(u,v) = (s,t)\,(x,y)^*(w,z) \subseteq \widehat{\theta}\,.$

[7] The length of a pair is the sum of the lengths of its components.

(b) Show that the *mirror* function $\rho\colon A^* \to A^*$:

$$\rho(a_1\, a_2 \cdots a_n) = a_n\, a_{n-1} \cdots a_1 \quad ,$$

is not a rational relation.

6. **Conjugacy.** Let $\mathsf{Conj}\colon A^* \to A^*$ be the relation which associates with every word w the set of its *conjugates*: $\mathsf{Conj}(w) = \{v\, u \mid u, v \in A^* \quad u\, v = w\}$.

(a) Show that if L is a rational language, then so is $\mathsf{Conj}(L)$.

(b) Give a transducer which associates with every word w of $\{a, b\}^*$ the word obtained by moving the first letter of w to its end.

(c) Compose this transducer with itself.

(d) Show that Conj is not a rational relation.

Notation Index

General Index