On Weakly Ambiguous Finite Transducers

Nicolae Santean and Sheng Yu

University of Western Ontario, London, ON N6A 5B8, Canada,
Department of Computer Science

Abstract. By weakly ambiguous (finite) transducers we mean those transducers that, although being ambiguous, may be viewed to be at arm's length from unambiguity. We define input-unambiguous (IU) and input-deterministic (ID) transducers, and transducers with finite codomain (FC). IU transductions are characterized by nondeterministic bimachines and ID transductions can be represented as a composition of sequential functions and finite substitutions. FC transductions are recognizable and can be expressed as finite unions of subsequential functions. We place these families along with uniformly ambiguous (UA) and finitely ambiguous (FA) transductions in a hierarchy of ambiguity. Finally, we show that restricted nondeterministic bimachines characterize FA transductions. Perhaps the most important aspect of this work consists in defining nondeterministic bimachines and describing their power by linking them with weakly ambiguous finite transducers (IU and FA).

1 Overview

Arguably one of the most intriguing machines that realize rational transductions are the bimachine, designed by Schutzenberger ([11]) and studied, among others, by Eilenberg who stated their importance in [3, §11.7, Theorem 7.1, p. 321]. A bimachine is a compact representation of the composition of a left and a right sequential transducer, and it characterizes the family of rational functions. A few variations of the original design have been studied in [9], where it has been shown that the scanning direction of its two reading heads does not matter. A natural question which has not been addressed so far is "what family of transductions are realized by bimachines that operate nondeterministically?". We show that these machines characterize the family of transductions that can be written as a composition of a rational function and a finite substitution. They are equivalent to the so-called input-unambiguous transducers (IU), which are close relatives of the classical unambiguous transducers. We also show that nondeterministic bimachines can "simulate" (i.e., they give a representation of) rational relations with finite codomain (FC). Surprisingly, we prove that FC transductions belong strictly to the family of recognizable relations and that they can be written as a finite union of subsequential functions. We notice that nondeterministic bimachines are a compact representation of the composition of a left sequential transducer and a right input-deterministic (ID) transducer - which is a close relative of the classical right sequential transducer. Finally, we define restricted nondeterministic bimachines to be those which do not reset themselves at each computation step. Surprisingly, we observe that this restriction increases their representation power, allowing them to characterize the entire family of finitely ambiguous (FA) rational relations. Basically, the reset/no-reset dichotomy reveals the difference between IU and FA families. Thus, by investigating the computational power of nondeterministic bimachines, we have been led to a study of various degrees of weak ambiguity in finite transducers.

The paper is structured as follows. In Section 2 we introduce transducers and ambiguity. We give a normalized form for IU transducers and we characterize ID transductions. Theorem 1 states the connection between IU and ID transductions by means of right sequential functions. In Section 3 we build a hierarchy of ambiguity by introducing FA, UA and FC transductions and by establishing their mutual relations. Since FC is a newly introduced family, we give a Mezei-like characterization of FC transductions, thus proving their recognizability (Theorem 2) and leading to a representation as a finite union of subsequential functions. Section 4 holds the most important results of the paper: theorems 3, 4 and 5. We define several types of nondeterministic bimachines, show that some types are equivalent and characterize the family of IU transductions, and reveal that restricted nondeterministic bimachines characterize FA transductions. All the proofs are omitted and can be found in [10].

2 Input-Unambiguous and Input-Deterministic Finite Transducers

In the following we assume known basic notions of automata theory ([5], [8], [13]). By DFA and NFA we understand deterministic and nondeterministic finite automata, and by ϵ-free NFA we understand NFA with no ϵ-transitions, where ϵ denotes the empty word.

By a finite transducer over the alphabets X and Y we understand a finite automaton over the product of free monoids $X^* \times Y^*$. In other words, a transducer is a finite automaton whose transition labels are elements of $X^* \times Y^*$, with the meaning that the first component of the label is an input word and the second component is an output word. It is well known that finite transducers realize `rational word relations` (see for example [8, §IV.1.2, p. 566]), denoted by $\mathrm{Rat}(X^* \times Y^*)$, or simply Rat when the alphabets are understood. By RatF we understand the family of rational functions.

Formally, a transducer is a tuple $T = (Q, X, Y, E, q_0, F)$, where Q is a set of states, X, Y are alphabets, q_0 is an initial state, F is a set of final states and E is a `finite` set of transitions which are elements of $Q \times X^* \times Y^* \times Q$. The transduction (binary word relation) realized by T will be denoted by $|T|$: $X^* \to Y^*$ and is defined similarly to the language accepted by an NFA. The transducer T is `normalized` if the following conditions hold:

1. $E \subseteq Q \times (X \cup \{\epsilon\}) \times (Y \cup \{\epsilon\}) \times Q$;
2. $F = \{q_f\}$, $q_f \neq q_0$;

3. $(p, x, \alpha, q) \in E \Rightarrow p \neq q_f, \;\; q \neq q_0$.

It is known that any rational transduction is realized by a normalized finite transducer and that any transducer can algorithmically be normalized.

By a `useful` state (or transition, path, loop, etc.) in a transducer we understand a state (or transition, path, loop, etc.) which is used in at least one successful computation. By an ϵ`-input loop` we understand a loop (in the transition graph) whose transitions have only ϵ-input labels.

The notion of ambiguity for automata (and transducers) relates to the number of possible successful computations performed by an automaton for a given input. For example, a DFA is unambiguous, whereas an NFA can have various degrees of ambiguity.

Definition 1. *An ϵ-NFA A is unambiguous (UNFA) if each word is the label of at most one successful computation in A.*

Let $T = (Q, X, Y, E, q_0, F)$ be a finite transducer. The `input automaton` of T is the finite automaton $A = (Q, X, \delta, q_0, F)$, where δ is given by

$$\forall x \in X^* : \;\; q \in \delta(p, x) \Leftrightarrow \exists \alpha \in Y^* : (p, x, \alpha, q) \in E, \;\; \text{where } p, q \in Q \;\; .$$

If the transducer is normalized, then its input automaton is an ϵ-NFA, otherwise it may be a lazy NFA (its transitions are labelled with words rather than letters or ϵ.).

Definition 2. *A finite transducer T is called* `input-unambiguous` *(IU, for short) if its input automaton is unambiguous (i.e., an UNFA).*

Notice that a transducer can still have different successful paths with same input labels and nevertheless be input-unambiguous. One such situation is depicted in Figure 1. Notice also the difference between this definition and the classical

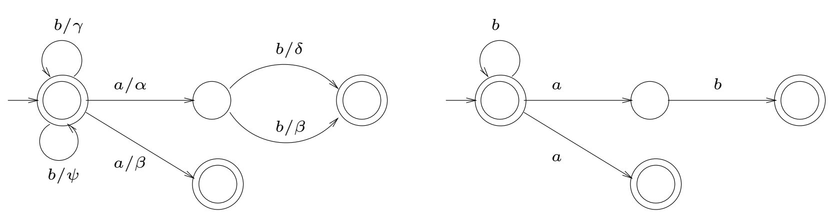

Fig. 1. An IU transducer and its input automaton.

definition of `unambiguous transducers` ([1, p. 114]).

Remark 1. In our formalism, we imply that an IU transducer cannot have useful ϵ-input loops, in the same way as an unambiguous automaton cannot have useful ϵ-loops.

An IU transduction is a transduction realized by an IU transducer. Given an arbitrary IU transducer, there exists an equivalent IU transducer in normal form, in the sense mentioned at the begining of Section 2. Indeed, the standard normalization algorithm (see for example [1, §III.6, Corollary 6, p. 79]) does not change the degree of ambiguity of a transducer.

We recall that a `trim transducer` has only useful states. Without loss of generality, we follow the convention that if the initial state of a transducer is also final then the pair (ϵ, ϵ) is realized by the transducer. This convention has a theoretical explanation which we choose to ignore here, due to its interference with the definition of ambiguity and normalization.

Lemma 1. *Any IU transduction* $\tau : X^* \to Y^*$ *with* $\tau(\epsilon) = \epsilon$ *or* $\tau(\epsilon) = \emptyset$ *is realized by a trim IU transducer* $T = (Q, X, Y, E, q_0, F)$ *which satisfy the following conditions:*

(i) $E \subset Q \times X \times Y^* \times Q$*;*
(ii) if $\tau(\epsilon) = \epsilon$ *then* $F = \{q_0, q_f\}$*, else* $F = \{q_f\}$*, and* $q_f \neq q_0$*;*
(iii) $(p, x, \alpha, q) \in E \Rightarrow q \neq q_f, p \neq q_0$.

One can notice that it is decidable whether a finite transducer is IU or not. The decision can be reduced to whether an ϵ-NFA is UNFA or not.

In the following we recall sequential transducers and functions in order to draw a parallel with ID transducers which will be defined in the following. A `(left) sequential transducer` is a tuple $T' = (Q, X, Y, \delta, \lambda, q_0)$, where Q, X and Y are as usual and $\delta : Q \times X \to Q$ and $\lambda : Q \times X \to Y^*$ are partial functions (transition and output functions) with a same domain ($dom(\delta) = dom(\lambda)$), that are extended in the usual way. This transducer is a particular finite transducer that has all its states final and has the transition set given by $E = \{(q, x, \lambda(q, x), \delta(q, x))/(q, x) \in dom(\delta)\}$. This type of transducers represents a subfamily of rational functions: sequential functions. A `right sequential transducer` is a sequential transducer that reads its input and writes its output from right to left. It is known that any rational function can be written as a composition of a left and a right sequential function ([1]).

Definition 3. *An* `input-deterministic` *(ID) transducer is a tuple* $T = (Q, X, Y, \delta, \omega, q_0)$ *where* X*,* Y *are alphabets,* Q *is a finite set of states, and*

$$\delta : Q \times X \to Q, \text{ and } \omega : Q \times X \to \mathcal{FP}(Y^*)$$

are partial functions with the same domain, denoting the transition and the output function. ($\mathcal{FP}(Y^*)$ *denotes all finite parts of* Y^**)*

In other words, an ID transducer is similar to a sequential transducer, with the exception that reading an input letter leads to a finite number of output choices. Notice that a transducer is ID if and only if its input automaton is deterministic – hence justifying its name. As usual we define the family of ID transductions to be the family of all transductions that are realized by ID transducers.

Lemma 2. *A transduction is ID if and only if it is the composition of a sequential transduction and a finite substitution.*

Theorem 1. *Let $\tau : X^* \to Y^*$ be a transduction with $\tau(\epsilon) = \epsilon$. Then τ is an IU transduction if and only if there exist a right sequential function $\mu : X^* \to Z^*$ and an ID transduction $\nu : Z^* \to Y^*$ such that $\tau = \nu \circ \mu$. Moreover, μ can be chosen to be total and length preserving.*

Intuitively, in the above decomposition the sequential transducer represents the set of unique successful paths of the unambiguous transducer, whereas the ID transducer represents the nondeterminism of the output process.

It is also worth mentioning that a transduction is IU if and only if it is the composition of a left sequential function and a "right" ID transducer, fact that can be proven similar to Theorem 1. Here, by a right ID transducer we understand a transducer that scans the input from right to left and writes the output from right to left as well. It is apparent by now the similarity between this characterization and the characterization of rational functions by right and left sequential functions.

3 A Hierarchy of Ambiguity

In order to place IU and ID transductions into a proper context, in the following we recall two known families of rational transductions: finitely and uniformly ambiguous.

Definition 4. *A rational transduction $\tau : X^* \to Y^*$ is* `finitely ambiguous` *(FA) if $\mid \tau(u) \mid < \aleph_0, \forall u \in X^*$. We say that τ is* `uniformly ambiguous` *(UA) if there is a constant N such that $\mid \tau(u) \mid < N, \forall u \in X^*$.*

These families of transductions have been studied and used in various application in the past ([4], [6]). For example, it is known that an UA rational transduction can be written as a finite union of rational functions ([6]), and one can easily decide whether a rational transduction is in FA (this is equivalent to detecting non-trivial ϵ-input loops in a finite transducer). However, we are not aware of whether it is decidable if a rational transduction is in UA or not. Next we aim at finding the relationship between all these families of rational word relations.

Corollary 1.

$$IU \subset FA \ .$$

This is a direct consequence of Remark 1: since an IU transducer has no ϵ-input loops, any input word can trigger a finite number of words to be written on the output tape. It affirms that the transductions realized by IU transducers are in FA, however they are not necessarily in UA. Indeed, the following example shows an IU transducer which realizes a transduction that is not uniformly ambiguous.

Example 1. The transducer in Figure 2 realizes the transduction τ given by:

$$\forall n \geq 1 : \ \tau(a^n) = \begin{cases} \bigcup_{i=1}^{n} \{x^i\}, & if \ n \ is \ even \\ \bigcup_{i=1}^{n} \{y^i\}, & otherwise \end{cases} ,$$

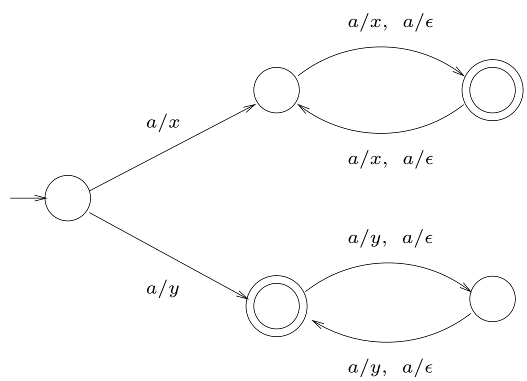

Fig. 2. An IU transducer whose transduction is not UA.

which clearly is not UA, however it is IU. On the other hand, not all rational transductions which are UA are necessarily IU. The transduction

$$\tau = \{(a^n, x^n)/n \geq 1\} \cup \{(a^n, y^n)/n \geq 1\} \qquad (1)$$

(with a, x, y different letters) is UA (notice that it is written as a union of two rational functions), however it is not IU. Indeed, a transducer T realizing τ must have two successful computations for each input word a^n: one outputting x^n and the other y^n, for all integers n. If these two successful computations coincide in the input automaton of T, then in T must exist a successful computation which "shuffles" x and y on the output tape, hence T cannot be IU.

Definition 5. *A rational transduction* $\tau : X^* \to Y^*$ *is with* `finite codomain` *(FC) if* $| \tau(X^*) | < \aleph_0$.

Obviously, it is decidable whether a rational transduction is in FC or not (it is equivalent to deciding whether the output automaton of a transducer accepts a finite language or not).

Lemma 3. *A rational transduction* $\tau : X^* \to Y^*$ *is in FC if and only if it can be written as*

$$\tau = \bigcup_{i \in I} [L_i \times R_i] \ ,$$

where I *is finite,* $\{L_i\}_{i \in I}$ *are disjoint regular languages and* $\{R_i\}_{i \in I}$ *are finite languages.*

One consequence of this lemma is the connection between transductions with finite codomain and subsequential transductions. Recall that a (`left`) `subsequential transducer` T' is a sequential transducer $T = (Q, X, Y, \delta, \lambda, q_0)$ (as defined in Section 2) together with a terminal output function $\rho : Q \to Y^*$, that realizes the rational function $| T' | (w) = | T | (w)\rho(\delta(q_0, w))$. It is known that there exist rational functions that can not be realized by either sequential or subsequential transducers. For more on the topic consult [1, §IV.2].

Corollary 2. *Any FC transduction can be written as a finite union of subsequential functions.*

In order to reveal the recognizability of FC transductions we recall that a recognizable set in a monoid is a set defined by an action over that monoid (see, for example, [8, p. 252]). Recall also that a subset of the direct product of two monoids (also a monoid) is recognizable if and only if it can be written as a finite union of blocks (a block is a direct product of two recognizable sets). This characterization is known as Mezei's characterization of recognizable sets in direct product monoids (see [3, Proposition 12.2, p. 68, and the note at p. 75]). Then, the recognizability of FC is a consequence of Lema 3. In the following, by *Rec* we understand the set of recognizable transductions over the alphabets X and Y, i.e., the family of recognizable subsets of $X^* \times Y^*$.

Theorem 2.

$$FC \subset Rec \cap IU \ .$$

Notice that obviously FC $\subset$ UA. Notice also that FC and the family of rational functions overlap, but are incomparable.

Remark 2. Although both FC and ID are included in IU, there is no relation of inclusion between FC and ID. For example, the transduction $\mu : \{a\}^* \to \{a\}^*$ given by

$$\forall n \geq 1 : \ \ \mu(a^n) = \bigcup_{i=1}^{n} \{a^i\}$$

is in ID but not in FC; whereas the transduction $\nu : \{a\}^* \to \{a, b\}^*$ given by

$$\forall n \geq 1 : \ \ \nu(a^n) = \begin{cases} a, & if \ n \ is \ even \\ b, & otherwise \end{cases}$$

is in FC (and in RatF, incidentally) but not in ID. Consequently, we may also infer that both FC and ID are `strictly` included in IU.

In Figure 3 we present a hierarchy describing different levels of ambiguity, where by dots we denote the areas where we have provided examples, including the following three:

$$FC \setminus (RatF \cup ID) : \ \ \tau_1(a^n) = \begin{cases} \{x, y\}, & if \ n \ is \ even \\ z, & otherwise \end{cases} \ ,$$

$$FA \setminus (UA \cup IU) : \ \ \tau_2(a^n) = \{\epsilon\} \cup \textstyle\bigcup_{i=1}^{n} x^i \cup \bigcup_{i=1}^{n} y^i \ ,$$

$$(UA \cap IU) \setminus (ID \cup RatF \cup FC) : \ \ \tau_3(a^n) = \begin{cases} \{x, y\}, & if \ n \ is \ even \\ z^n, & otherwise \end{cases} \ .$$

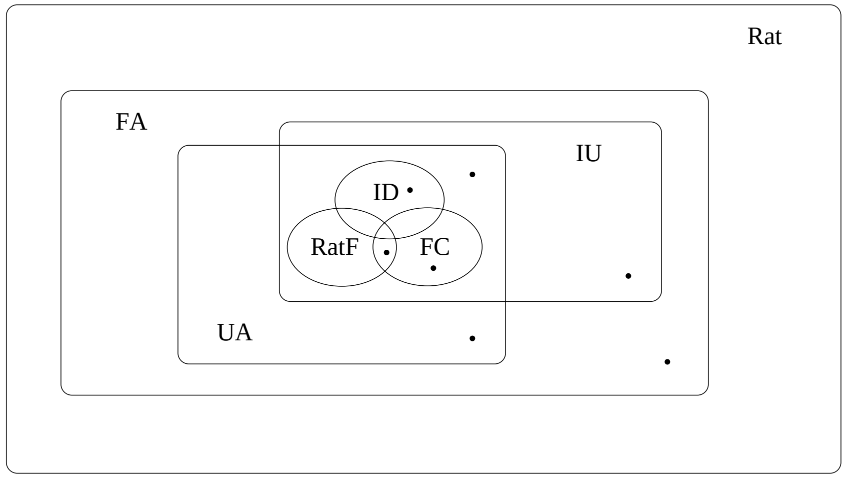

Fig. 3. Different degrees of ambiguity (dots represent examples).

4 Nondeterministic Bimachines

In the following we consider all input-unambiguous transducers to be trim and normalized according to Lemma 1. We are now aiming at giving a bimachine-characterization of IU.

Definition 6. *A* `bimachine` $B = (Q, P, X, Y, \delta_Q, \delta_P, q_0, p_0, \omega)$ *over* X *and* Y *is composed of*

two finite sets of states Q *and* P*,*
a finite input alphabet X *and a finite output alphabet* Y*,*
two partial next state functions

$$\delta_Q : Q \times X \to Q \text{ and } \delta_P : X \times P \to P \ ,$$

two initial states $q_0 \in Q$ *and* $p_0 \in P$*,*
and a partial output function $\omega : Q \times X \times P \to Y^*$*.*

The next-state functions are extended to operate on words as follows:

– $\forall q \in Q$ and $p \in P : \delta_Q(q, \epsilon) = q$ and $\delta_P(\epsilon, p) = p$;
– $\forall q \in Q, p \in P, a \in X$ and $w \in X^+$:

$$\delta_Q(q, wa) = \delta_Q(\delta_Q(q, w), a) \text{ and } \delta_P(aw, p) = \delta_P(a, \delta_P(w, p)).$$

Notice that function δ_P "reads" its argument word in reverse. We consider a similar extension of the output function:

– $\forall q \in Q$ and $p \in P : \omega(q, \epsilon, p) = \epsilon$;

– $\forall q \in Q, p \in P, a \in X$ and $w \in X^+$:

$$\omega(q, wa, p) = \omega(q, w, \delta_P(a, p))\omega(\delta_Q(q, w), a, p).$$

The partial word function realized by B is a function $f_B : X^* \to Y^*$, defined by $f_B(w) = \omega(q_o, w, p_0)$ if ω is defined in (q_0, w, p_0) and is undefined otherwise. Notice that $f_B(\epsilon) = \epsilon$ for any bimachine B. In essence, a bimachine is composed of two partial automata without final states (more precisely, all states act as final) and an output function. Indeed, (Q, X, δ_Q, q_0) will denote the `left automaton` of B and (P, X, δ_P, p_0) its `right automaton`.

Bimachines are of great theoretical importance since they are specifically designed to characterize the family of rational word functions. To our knowledge, so far there has been no attempts to study nondeterministic bimachines. We distinguish 3 components of a bimachine which are candidate to nondeterminism: the left and right automata and the output function. According to this, we define the following new types of bimachines:

1. `FNObm` : with finitely nondeterministic output (at each "step" the bimachine nondeterministically writes a word on the output tape, choosing from a finite set of choices);
2. `NTbm` : with nondeterministic transitions (the two underlying automata are nondeterministic: ϵ-NFA);
3. `LNTbm` : with left nondeterministic transitions (only the "left automaton" is nondeterministic);
4. `RNTbm` : with right nondeterministic transitions (only the "right automaton" is nondeterministic);
5. `NTObm` : with both nondeterministic transitions and finitely nondeterministic output;

and we denote by FNO, NT, LNT, etc. the families of transductions realized by these types of bimachines.

It is important to observe that at each computation step of an NTbm B, both the left and the right automata of B are "reset" to their initial state. This point is made clear in Figure 4. While reading w_1, the left automaton reaches the state q, through the computation(path) labelled w_1. However, in the next computation step, the left automaton reads $w_1 a$ and performs the computation labelled $w_1 a$ that may not overlap with the previous computation (more precisely, the computation labelled $w_1 a$ is not necessarily prefixed by the computation labelled w_1). This is due to the fact that the left automaton is reset to the initial state before reading $w_1 a$ (it does not continue the computation from q while reading a).

Theorem 3.

$$FNO = NT = LNT = RNT = NTO \ .$$

In other words, it does not matter which component of the bimachine is nondeterministic. For this reason, we are allowed to employ the term `nondeterministic bimachine` in a generic sense.

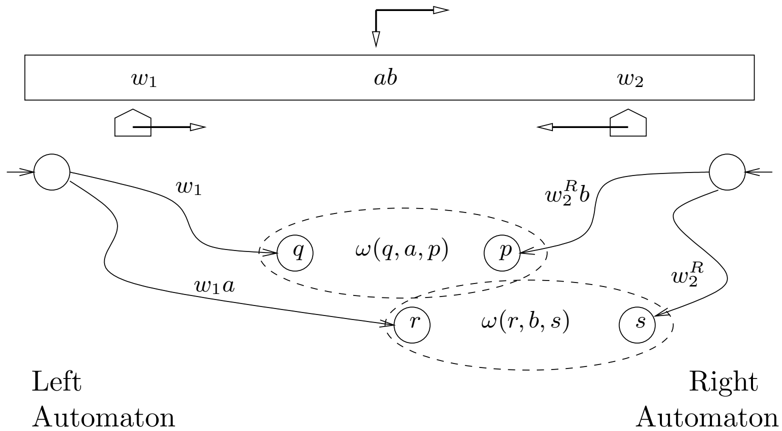

Fig. 4. NTbm behavior: each computation step involves a "reset".

It has been shown in [9] that the scanning direction of the reading heads of a (deterministic) bimachine does not matter. It is natural to question whether this property still holds for nondeterministic bimachines.

Corollary 3. *The parsing direction of the reading heads of a nondeterministic bimachine does not matter.*

The same statement also applies to restricted nondeterministic bimachines - defined later. It tells that convergent, left sequential, right sequential, and divergent nondeterministic bimachines all have equal power. This is a consequence of Theorem 3: one may use FNO bimachines and adapt the proof in [9, T.16, p. 135] to the nondeterministic case.

We are now ready to state one of the main results of this paper, namely a bimachine characterization of IU rational transductions.

Theorem 4. *A transduction τ with $\tau(\epsilon) = \epsilon$ is IU rational* `if and only if` *it is realized by a nondeterministic bimachine.*

Consequence of Lemma 2 and Theorem 1 we obtain another characterization of IU transductions, that by Theorem 4 becomes a characterization of nondeterministic bimachines as well:

Corollary 4. *A transduction $\tau : X^* \to Y^*$ is IU if and only if there exists a rational function $\mu : X^* \to Z^*$ and a finite substitution $\sigma : Z^* \to \mathcal{FP}(Y^*)$ such that $\tau = \sigma \circ \mu$.*

Notice that it is decidable whether a nondeterministic bimachine is single-valued (realizes a rational function). Indeed, one can first construct an equivalent IU transducer whose functionality can be decided ([12], [2]). Notice also that the number of outputs for a given input of an IU transduction is a linear function of the length of the input and the length of any output is also a linear function of the length of the input. The converse does not hold, as the transduction (1) in Section 3, Example 1 is not IU, however it verifies these conditions. Finally, a surprising consequence of Corollary 4 and Theorem 2 is that any FC transduction can be represented by a composition of a rational function and a finite substitution as well.

So far we have introduced nondeterministic bimachines with a special behavior: at each computation step, these bimachines perform a "reset", i.e., they set their underlying automata to be in initial state. Then a natural question occurs, that is, "what would happen if we inhibit the reset?". This leads to the definition of another type of nondeterministic bimachine: a `restricted nondeterministic bimachine`. At each step, these bimachines are forced to continue their computation from the states reached at the previous step (nevertheless, they remain nondeterministic).

Definition 7. *A restricted nondeterministic bimachine (RNTbm) is a bimachine with nondeterministic transitions (NTbm) and multiple initial states* $B = (Q, P, X, Y, \delta_Q, \delta_P, I_Q, I_P, \omega)$*, where the output function is extended as follows:*

- $\forall q \in Q, p \in P: \ \ \omega(q, \epsilon, p) = \{\epsilon\}$*;*

- $\forall w = a_1 ... a_n \in X^+$ *(where* $\forall i \in \{1, ..., n\} : a_i \in X$*),*
  $\forall q_0 \in I_Q, p_0 \in I_P$*,* $\omega(q_0, w, p_0)$ *is given by:*

$$\{ \ \ \omega(q_0, a_1, p_{n-1})\omega(q_1, a_2, p_{n-2})...\omega(q_{n-2}, a_{n-1}, p_1)\omega(q_{n-1}, a_n, p_0) \ \ / \\ q_1 \in \delta_Q^*(q_0, a_1), ..., q_{n-1} \in \delta_Q^*(q_{n-2}, a_{n-1}), \\ p_1 \in \delta_P^*(a_n, p_0), ..., p_{n-1} \in \delta_P^*(a_2, p_{n-2}) \ \ \}$$

Notice that by this behavior, the bimachine still operates nondeterministically. However, the current states of its automata depend on the previous current states. Surprisingly, although this seems like a restriction, RNTbm's have a greater power than NTbm's. Notice also that we allow multiple initial states - for improving the formalism. At the beginning of the operation, a RNT bimachine sets itself nondeterministically into two initial states corresponding to its left and right automata.

Theorem 5. *A transduction* τ *with* $\tau(\epsilon) = \epsilon$ *is in FA if and only if it is realized by a RNTbm.*

This theorem together with Theorem 4 completes the characterization of nondeterministic bimachines: they realize either IU or FA rational transductions, with respect to whether a reset is or not in place. Notice in Figure 3 the gap between deterministic bimachines (RatF) and nondeterministic ones (IU, FA).

5 Conclusion and Further Work

The goal of this paper has been twofold: to introduce nondeterministic bimachines and to study weakly ambiguous finite transducers. Nondeterministic bimachines can realize FC relations; however, they can do better than that: they exactly represent the family of transductions that are the composition of rational functions and finite substitutions. The transducer counterpart of these machines is the input-unambiguous transducer, which is a slight variation of the classical notion of unambiguous transducer. FC relations are recognizable and they have a particular "Mezei representation", as a finite union of blocks with certain properties: their left components are disjoint and their right ones are finite. This leads in a natural way to the representation of FC relations as a finite union of subsequential functions - notice the parallel with the uniformly ambiguous rational relations, that are finite unions of rational functions. Nondeterministic bimachines can work in two "modes": with or without reset. We have proven that suppressing the reset in between computation steps increases their power: they now characterize the family of finitely ambiguous transductions. Finally, we believe that all major rational families of transductions have a "bimachine" counterpart. In particular, we leave for immediate work the study of "ϵ-RNT" bimachines (i.e., RNT bimachines with ϵ-advancement) that we believe characterize the entire family of rational relations.

References

1. Jean Berstel. *Transductions and Context-Free Languages*. B. G. Teubner, Stuttgart, 1979.
2. Meera Blattner and Tom Head. Single Valued a-Transducers. *Journal of Computer and System Sciences*, 15(3):310–327, 1977.
3. Samuel Eilenberg. *Automata, Languages and Machines*, volume A. Academic Press, Inc., Orlando, FL, 1974.
4. Eitan M. Gurari and Oscar H. Ibarra. A Note on Finite-Valued and Finitely Ambiguous Transducers. *Mathematical Systems Theory*, 16(1):61–66, 1983.
5. John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation - 1st edition*. Addison-Wesley Longman Publishing Co. Inc., Boston, MA, 1979.
6. Emmanuel Roche and Yves Schabes. Introduction to Finite-State Devices in Natural Language Processing. Technical Report TR-96-13, Mitsubishi Electric Research Laboratories, 1996.
7. Grzegorz Rozenberg and Arto Salomaa. *Handbook of Formal Languages*. Springer Verlag, Berlin Heidelberg New York, 1997.
8. Jacques Sakarovitch. *Éléments de Théorie des Automates*. Vuibert Informatique, Paris, 2003.
9. Nicolae Santean. Bimachines and Structurally-Reversed Automata. *Journal of Automata, Languages and Combinatorics*, 9(1):121–146, 2004.
10. Nicolae Santean and Sheng Yu. Nondeterministic Bimachines and Rational Relations with Finite Codomain. Technical Report 649, ISBN-10: 0-7714-2552-X, ISBN-13: 978-0-7714-2552-X, University of Western Ontario, 2005.
11. Marcel-Paul Schützenberger. A Remark on Finite Transducers. *Information and Control*, 4(2-3):185–196, 1961.
12. Marcel-Paul Schützenberger. Sur les Relations Rationelles. *Lecture Notes in Computer Science*, 33:209–213, 1975.
13. Sheng Yu. Regular Languages. *In [7]*, 1:41–110, 1997.