Nominal deterministic Muller automata

Vincenzo Ciancia, Matteo Sammartino

Abstract. We define a class of stream languages, and the corresponding automata, that operate on infinite alphabets. The automata used for recognition are a generalisation of deterministic Muller automata to the case of infinite alphabets. Closure under complement, union and intersection is retained from the classical case.

1 Introduction

2 Background

Notation. As a matter of notation, for X, Y sets, we let $f\colon X \to Y$ be a total function from X to Y, $f\colon X \rightarrowtail Y$ be a total injective function and $f\colon X \rightharpoonup Y$ a partial function. Given a (partial) injective function f, we write $f^{-1}\colon Y \rightharpoonup X$ for the partial inverse function $\{(y,x) \mid f(x) = y\}$. We let $f|_{X'}$, with $X' \subseteq X$, be the domain restriction of f to X'.

Todo 1 *Introduce nominal sets and the notation $Orb(X)$ for the set of orbits of X.*

3 Nominal regular ω-languages

Traditionally, automata can be deterministic or non-deterministic. In order to extend previous results on closure under complementation and decidability, one needs to work with deterministic structures; therefore, in this paper, we introduce directly the deterministic structures.

Definition 1. *A* nominal deterministic Muller automaton *(nDMA) is a tuple $(Q, \longrightarrow, q_0, \mathcal{A})$ where:*

- *Q is an orbit-finite nominal set of* states*;*
- *$q_0 \in Q$ is the* initial state*;*
- *$\mathcal{A} \subseteq \mathcal{P}(Orb(Q))$ is a set of sets of orbits, intended to be used as an accepting condition in the style of Muller automata.*
- *$\longrightarrow$ is the* transition relation*, made up of triples $q_1 \xrightarrow{l} q_2$, having* source *q_1,* target *q_2,* label *$l \in \mathcal{N}$;*
- *the transition relation is* deterministic*, that is, for each $q \in Q$ and label $l \in \mathcal{N}$ there is exactly one transition with source q and label l.*

4 Finite automata

Definition 2. *An* history-dependent deterministic Muller automaton *(hDMA) is a tuple* $(Q, |-|, q_0, \rho_0, \longrightarrow, \mathcal{A})$ *where:*

- Q *is a finite set of* states*;*
- *for* $q \in Q$, $|q|$ *is a finite set of* local names *(or* registers*) of state* q*;*
- $q_0 \in Q$ *is the* initial state*;*
- $\rho_0 : |q_0| \to \mathcal{N}$ *is the* initial assignment*;*
- $\mathcal{A} \subseteq \mathcal{P}(Q)$ *is the* accepting condition*, in the style of* Muller automata*;*
- $\longrightarrow$ *is the* transition relation*, made up of quadruples* $q_1 \xrightarrow[\sigma]{l} q_2$*, having* source q_1*,* target q_2*, label* $l \in |q_1| \uplus \{\star\}$*, and* history $\sigma : |q_2| \rightarrowtail |q_1| \uplus \{l\}$*;*
- *the transition relation is* deterministic *in the following sense: for each* $q_1 \in Q$*, there is exactly one transition with source* q_1 *and label* $\star$*, and exactly one transition with source* q_1 *and label* x *for each* $x \in |q_1|$*.*

In the following we fix a HDMA $A = (Q, |-|, q_0, \rho_0, \longrightarrow, \mathcal{A})$. Acceptance of an word $\alpha \in \mathcal{N}^\omega$ is defined in terms of the *configuration graph* of A.

Definition 3. *The set* $\mathcal{C}$ *of* configurations *of* A *consists of the pairs* (q, ρ) *such that* $q \in Q$ *and* $\rho : |q| \to \mathcal{N}$*.*

Definition 4. *The* configuration graph *of* A *is a transition relation over triples* $(q_1, \rho_1) \xrightarrow{a} (q_2, \rho_2)$ *where the source and destination are configurations, and* $a \in \mathcal{N}$*. There is one such transition if and only if there is a transition* $q_1 \xrightarrow[\sigma]{l} q_2$ *in* A *and either of the following happens:*

- $l \in |q_1|$*,* $\rho_1(l) = a$*, and* $\rho_2 = \rho_1 \circ \sigma$*;*
- $l = \star$*,* $a \notin \Im(\rho_1)$*,* $\rho_2 = (\rho_1 \circ \sigma)[{}^{a}/_{\sigma^{-1}(\star)}]$*.*

The definition deserves some explanation. Fix a configuration (q_1, ρ_1). Say that name $a \in \mathcal{N}$ is *assigned to* the register $i \in |q_1|$ if $\rho_1(i) = a$. When a is not assigned to any register, it is fresh for a given configuration. Then the transition $q_1 \xrightarrow[\sigma]{l} q_2$, under the assignment ρ_1, consumes a symbol as follows: either $l \in |q_1|$ and a is the name assigned to register l, or l is $\star$ and a is fresh. The destination assignment ρ_2 is defined using σ as a binding between local registers of q_2 and local registers of q_1, therefore composing σ with ρ_1 and eventually associating a freshly received name, whenever $\star$ is in the image of σ.

We write $(q_1, \rho_1) \stackrel{v}{\Longrightarrow} (q_2, \rho_2)$ for the path in the configuration graph that spells v. Notice that, being A deterministic, there can only be one such path. Given a sequence P of transitions in A, we write $(q_1, \rho_1) \stackrel{v}{\Longrightarrow}_P (q_2, \rho_2)$ whenever $(q_1, \rho_1) \stackrel{v}{\Longrightarrow} (q_2, \rho_2)$ and such path is yielded by P.

Now we analyze properties of *loops*, i.e. sequences of transitions whose initial and final state coincide. Consider a loop $L := p_0 \xrightarrow[\sigma_0]{l_0} p_1 \xrightarrow[\sigma_1]{l_1} \dots \xrightarrow[\sigma_{n-1}]{l_{n-1}} p_0$ in A. Let $\underline{i}$ denote $i \mod n$. Let $\widehat{\sigma}_i : |p_{\underline{i+1}}| \rightharpoonup |p_i|$ be the partial functions given by

$$\widehat{\sigma}_i := \sigma_i \setminus \{(x, y) \in \sigma_i \mid y = \star\} \qquad (i = 1, \dots, n)$$

Intuitively, these are the maps between local names induced by σ_i ignoring allocations. Let $\widehat{\sigma}\colon |p_0| \rightharpoonup |p_0|$ be their composition $\widehat{\sigma}_0 \circ \widehat{\sigma}_1 \cdots \circ \widehat{\sigma}_n$. We define the set I as the greatest subset of $dom(\widehat{\sigma})$ such that

$$\widehat{\sigma}(I) = I \ ,$$

i.e. I contains those names that are just permuted along the loop. We denote by T all the other names, namely

$$T := |p_0| \setminus I \ .$$

The following lemma says that names stored in T are eventually forgotten.

Lemma 1. *Given any $x \in T$, let $\{x_j\}_{j \in J_x}$ be the smallest sequence that satisfies the following conditions*

$$x_0 = x \qquad x_{i+1} = \sigma_{\underline{i}}^{-1}(x_i)$$

where $i+1 \in J_x$ only if $\sigma_{\underline{i}}^{-1}(x_i)$ is defined. Then J_x has finite cardinality.

Proof. First of all, observe that this sequence is such that $x_{kn} \neq x_{k'n}$, for all $k, k' \geq 0$ such that $k \neq k'$. In fact, suppose there are $x_{kn} = x_{k'n}$, with $k < k'$. Then we should have $x_{kn-1} = x_{k'n-1}$, because σ_n is injective. In general, $x_{kn-l} = x_{k'n-l}$, for $0 \leq l \leq kn$, therefore $x = x_0 = x_{(k'-k)n}$, which implies $x \in I$, against the hypothesis $x \in T$.

Caratterizzare I anche come l'insieme dei registri che prima o poi vengono mappati su loro stessi?

Now, suppose that $J_x = \mathbb{N}$. Then we would have an infinite subsequence $\{x_{kn}\}_{k \in \mathbb{N}}$ of pairwise distinct names that belong to $|p_0|$, but $|q_0|$ is finite, a contradiction. $\square$

Lemma 2. *Given a path, there is always a word that follows that path.*

Lemma 3. *Given any $\rho\colon |p_0| \to \mathcal{N}$:*

1. *there is $\theta \geq 1$ such that, for all $v_1, \ldots, v_\theta$ such that*

$$(p_0, \rho) \xRightarrow{v_1}_L (p_0, \rho_1) \xRightarrow{v_2}_L \ldots \xRightarrow{v_\theta}_L (p_0, \rho_\theta)$$

   we have $\rho_\theta|_I = \rho|_I$;
2. *there is $\epsilon \geq 1$ such that, for all $\gamma \geq \epsilon$, there are $v_1, \ldots, v_\gamma$ such that*

$$(p_0, \rho) \xRightarrow{v_1}_L (p_0, \rho_1) \xRightarrow{v_2}_L \ldots \xRightarrow{v_\gamma}_L (p_0, \rho_\gamma);$$

   and $\rho_0(T) \cap \rho_\gamma(T) = \varnothing$. Fix: $\rho_0(|p_0|) \cap \rho_\gamma(T) = \varnothing$.
3. *there is ζ such that, for any $\rho'\colon |p_0| \to \mathcal{N}$ with $\rho(T) \cap \rho'(T) = \varnothing$, there are $v_1, \ldots, v_\zeta$ such that*

$$(p_0, \rho) \xRightarrow{v_1}_L (p_0, \rho_1) \xRightarrow{v_2}_L \ldots \xRightarrow{v_\zeta}_L (p_0, \rho_\zeta)$$

   and $\rho_\zeta|_T = \rho'|_T$.

Proof.

1. Notice that $\widehat{\sigma}|_I$ is a permutation, so by Langrange's theorem there is θ such that $\widehat{\sigma}|_I^\theta = id_I$. The claim follows from $\rho_\theta|_I = \rho|_I \circ \widehat{\sigma}|_I^\theta = \rho|_I$.
2. Let $\mathcal{J}$ be
$$\mathcal{J} := \max\{|J_x| \mid x \in T\} + 1.$$
This gives the number of transitions it takes to forget all the names stored in T. Let ϵ be $\lceil \frac{\mathcal{J}}{n} \rceil$. For any $\gamma \geq \epsilon$, we can choose $v_1, \dots, v_\gamma$ as any γ-tuple of words that are recognized by the loop and such that, whenever $l_j = \star$, then $(v_i)_j$ is different from $\rho(|p_0|)$ and all the previous symbols in $v_1, \dots, v_i$, for all $i = 1, \dots, \gamma$ and $j = 1, \dots, n$. The final assignment ρ_γ clearly satisfies the statement: none of the names in $\rho_\gamma(T)$ come from old names in $\rho(T)$, as they are all forgotten and replaced by fresh ones.
3. For each name $x \in T$ define a tuple (x, i, j) where i is the index of the transition where x is allocated and j is the number of loop traversals needed to allocate it, i.e. j is the smallest integer such that there are $x_{jn}, \dots, x_1$ defined as follows
$$x_{jn} = x \qquad \sigma_{\underline{k+1}}(x_{k+1}) = x_k \qquad \sigma_i(x_1) = \star$$
Let X be the set of such tuples and let $\zeta := \max\{j \mid (x, i, j) \in X\}$. Then we can construct $v_1, \dots, v_\zeta$ as follows
$$(v_k)_i := \begin{cases} y \text{ fresh} & l_i = \star \wedge i \notin \pi_2(X) \\ \rho'(x) & (x, i, \zeta - k + 1)^1 \in X \\ \rho(l_0) & l_0 \neq \star \\ \rho_{k-1}(l_i) & i > 0, l_i \neq \star \end{cases}$$
Lemma 4. *Given a transition*
$$(p_1, \rho_1) \xrightarrow{a} (p_2, \rho_2)$$
we have $\rho_2(|p_2|) \subseteq \rho(|p_1|) \cup \{a\}$.

where by y fresh we mean different from elements of $\rho(|p_0|) \cup \rho'(|p_0|)$ and previous symbols in $v_1, \dots, v_k$. The second case is allowed by $\rho(T) \cap \rho'(T) = \varnothing$ and lemma 4.Then ρ_ζ satisfies the statement by construction.

Dire meglio!

□

Proposition 1. *For any* $\hat{\rho} \colon |p_0| \to \mathcal{N}$ *there are* $v_1, \dots, v_n$ *such that*
$$(p_0, \hat{\rho}) \xRightarrow{v_1}_L (p_0, \hat{\rho}_1) \xRightarrow{v_2}_L \cdots \xRightarrow{v_n}_L (p_0, \hat{\rho}) \ .$$

Proof. We can take any path of the form
$$(p_0, \hat{\rho}) \xRightarrow{v_1}_L (p_0, \hat{\rho}_1) \xRightarrow{v_2}_L \cdots \xRightarrow{v_\gamma}_L (p_0, \hat{\rho}_\gamma) \xRightarrow{v_{\gamma+1}}_L \cdots \xRightarrow{v_{\gamma+\zeta}}_L (p_0, \hat{\rho}_{\gamma+\zeta})$$

where the subpath from $(p_0, \hat{\rho})$ to $(p_0, \hat{\rho}_\gamma)$ is given by 2 of lemma 3 and the remaining subpath is given by 3 of the same lemma, with $\rho = \hat{\rho}_\gamma$ and $\rho' = \hat{\rho}$. The only constraint about γ is that there should be a positive integer λ such that $\gamma + \zeta = \lambda\theta$, where θ is given by 1 of lemma 3. Thanks to the lemma we have $\hat{\rho}_{\gamma+\zeta}|_T = \rho|_T$ and $\hat{\rho}_{\gamma+\zeta}|_I = \rho|_I$ which, together with $I \cup T = |p_0|$, imply $\hat{\rho}_{\gamma+\zeta} = \hat{\rho}$. □

Theorem 1. *Every non-empty language $\mathcal{L}$ recognized by a HDMA A has an ultimately periodic fragment.*

Proof. Take any string α recognized by A and let $I = Inf(q_0, \alpha)$. A run for α in the configuration graph must begin with

Spiegare meglio perchè “must”?

$$(q_0, \rho_0) \xRightarrow{u} (q_1, \rho_1) \xRightarrow{v}_P (q_1, \rho_2)$$

where $q_1 \in I$ and $(q_1, \rho_1) \xRightarrow{v}_P (q_2, \rho_2)$ is a path, induced by some sequence of transitions P in A, that goes through all the states in I. Since P is a loop, we can replace the second path with a new one given by proposition 1

$$(q_0, \rho_0) \xRightarrow{u} (q_1, \rho_1) \xRightarrow{v_1}_P \cdots \xRightarrow{v_n}_P (q_1, \rho_1) \ .$$

This subpath can be traversed any number of times, so we have $u(v_1 \ldots v_n)^\omega \in \mathcal{L}$. □

5 Boolean operations

6 Ultimately-periodic determinacy

7 Nominal regularity of looping fragments

8 Decidability

Todo 2 *Quick introduction to [1].*

References

1. Ciancia, V., Venema, Y.: Stream automata are coalgebras. In: Coalgebraic Methods in Computer Science. Volume 7399 of LNCS. Springer (2012) 90–108