Mathematical Approach in Automata and Automata Association

Sérgio Henrique Maciel
sergio.hm@aluno.ifsc.edu.br
November 2020

Abstract

The transition structure of an automaton can be used to create a natural topology to the set of states of an automaton, generating, this way, a topological space. Probabilistic automata can also be modeled in terms of measure theory. A system of many automata would be reduced to simple mathematical structures and analyzed by a topological point of view.

Introduction

The Automata Theory advanced considerably after 50s, based in the formalization of languages and the way automata work, allowing a good understanding about the states and transition chains of a generic automaton. A second study line has been developed based in topological principles behind all the formal structure of an automaton. In opposition of the main lines, we could be not interested in analyze the behavior of an Automaton from its language, but in the set of states itself and relations between different set of states, it is, different automata.

We can get it clearer from a fresh notion about what is an automaton. One could think about automaton as a thing with states, some ability to receive and read messages, information, and change its states based in the known information. A machine in a production line, for example, can be described as an automaton, having different states which could be information about its position an currently actions or tasks, with some sensors sending electrical signs to tell the machine when to change its state and to which state go now.

The automaton notion also applies to computer softwares. States don’t need to be physical actions or positions and the messages don’t need to be sensors signs. There are actually even mechanical automata, with no electrical signs or similar thing.

The traditional study of an automata is focused in the set of possible messages it supports, its properties and how it defines the automaton as it is. But the present work has as object, the set of states, using mainly topological
principles from mathematical theory. Also, we are interested in how different automata work together.

In the first part, we are going to introduce the mathematical tools used along the paper. This includes fast notes about topology, a bit of measure theory and probability, as well as a brief introduction to graph theory, not going further than we would need to understand the results. This part is not supposed to replace a course in the quoted areas, we just want to make a very short review or offer a easy and fast way to consult the topics without needing to reopen a big book.

In the second part, we start to applying the topological theory in automaton theory, providing definitions about interesting automata properties and its relation with topological properties that the set of states can have. We will have a good amount of tools to analyze the structure of different automata.

In the third part we will briefly discuss the special cases of finite automata. We are going to spend some time looking for finite automata definitions and some of the already defined properties in this case, as well as using probability theory to create a more sophisticated and much more useful definition of Probabilistic Automaton, which we will be able to use in the fourth part.

In the forth and last part, our focus is in systems which involve multiple automata simultaneously. We want to build a way to identify how they can affect each other and provide tools to describe these interactions and the global result of them.

1 A Review of the Mathematical Theory involved

This section is supposed to make some exposure, not in a depth way, about the mathematical topics this article uses. Most of this part is intended to present topics in Topology, but there will fundamental points in Probability Theory and, hence, Measure Theory.

1.1 Topology

Topology studies intrinsic properties of sets. We won’t care about how the set is immersed in some other space, but specifically about properties which are invariant under referential changes of view.

**Definition 1.1.** Let $X$ be a set and $\mathcal{I}$ a family of subsets of $X$, such that

1. $\emptyset, X \in \mathcal{I}$;
2. The union of an arbitrary number of members in $\mathcal{I}$ is also in $\mathcal{I}$;
3. The intersection of an arbitrary number of members in $\mathcal{I}$ is also in $\mathcal{I}$.

The pair $(X, \mathcal{I})$ is called **Topological Space**.

The members of $\mathcal{I}$ are called **open sets**.
Example 1.2. Given a set $X$, $\mathcal{T}_{\text{ind}} = \{\emptyset, X\}$ is a topology in $X$. We call it Indiscrete Topology.

Example 1.3. Given a set $X$, $\mathcal{T}_{\text{dis}} = \mathcal{P}(X)$ is a topology in $X$, such that $\mathcal{P}(X)$ represents the power set of $X$, it is, the family of all subsets of $X$. We call this topology the Discrete Topology.

Example 1.4. Given the set $X = \{a, b, c\}$, $\mathcal{T}_1 = \{\emptyset, X, \{a\}\}$ and $\mathcal{T}_2 = \{\emptyset, X, \{a\}, \{b\}, \{a, b\}\}$ are typologies in $X$. The reader can verify the three conditions are satisfied.

Definition 1.5. Let $(X, \mathcal{T}_X)$ e $(Y, \mathcal{T}_Y)$ be two topological spaces. A function $f : X \to Y$ is continuous if $\forall U \in \mathcal{T}_Y$, $f^{-1}(U) \in \mathcal{T}_X$.

Example 1.6. A function $f : (X, \mathcal{T}_{\text{disc}}) \to (Y, \mathcal{T}_Y)$ is always continuous. Indeed, if $V \subset Y$, $f^{-1}(V) \in \mathcal{T}_{\text{disc}}$ since the topology is discrete and $f^{-1}(V)$ will be a subset of $X$.

Definition 1.7. Let $(X, \mathcal{T}_X)$ and $(Y, \mathcal{T}_Y)$ be two topological spaces. We define the Product Topology of the set $X \times Y$ as $\mathcal{T}_{\text{prod}} = \{U \times V \mid U \in \mathcal{T}_X, V \in \mathcal{T}_Y\}$.

Definition 1.8. A function $f : X \to Y$ between the topological spaces $X, Y$ is said to be a homeomorphism if, and only if, $f$ is bijective, continuous and $f^{-1}$ is also continuous.

Existing a homeomorphism between $X$ and $Y$, we say $X$ is homeomorphic to $Y$ and we write it as $X \cong Y$.

Example 1.9. Every opened interval $(a, b) \subset \mathbb{R}$ is homeomorphic to $\mathbb{R}$.

An intuitive way to think about homeomorphism is to look at them as a qualitative description which is well used in situations where is not needed a lot of rigor. In this description, we can think two spaces are homeomorphic if is possible to transform a space in another only with continuous deformations, it means, without cutting or gluing parts of the space. But you must notice it is very hard to use this interpretation in discrete spaces.

Definition 1.10. A topological space $(X, \mathcal{T}_X)$ is said to be Disconnected if there are $U, V \subset X$ open sets and not empty, such that $X = U \cup V$ and $U \cap V = \emptyset$. A topological space is said to be Connected if it is not disconnected.

Example 1.11. If $\mathbb{R}$ has the usual topology, which we didn’t define but intuitively is how we work naturally in euclidean space, the unique connected subsets are the open intervals $(a, b)$ for $a, b \in \mathbb{R}$.

Example 1.12. $\mathbb{Q}$ is a disconnected subset in $\mathbb{R}$, the sets $A = (-\infty, \sqrt{2}) \cap \mathbb{Q}, B = (\sqrt{2}, +\infty) \cap \mathbb{Q}$ are disjoint and the union $A \cup B$ contains $\mathbb{Q}$.

Definition 1.13. Let $X$ be a set. A Closure operator in $X$ is a function $C : \mathcal{P}(X) \to \mathcal{P}(X)$ which satisfies, for all $S_1, S_2 \subset X$,

- $S_1 \subseteq C(S_1)$;
- $S_1 \subseteq S_2 \Rightarrow C(S_1) \subseteq C(S_2)$;
- $C(C(S_1)) = C(S_1)$.
A closure operator induces a topology in the set which it is defined. The resultant sets of the application of a closure operator are closed sets, its complements are open sets and define a topology.

For a further reading with more details and complements, the reader can search for many books and materials in Topology branch, for example, [4]. The material provides a better explanation which can be useful for a deeper understanding of the work.

1.2 Measure and Probability Theory

The measure theory looks for ways to associate values to subsets of a set, following some definitions. It can be used to define volume, for example. On the other hand, Probability Theory uses the measure theory ideas to associate probabilistic values to events.

Definition 1.14. A $\sigma$-algebra of a set $X$ is a collection $\mathcal{A} = \{A_1, A_2, \ldots\}$ of subsets of $X$, such that

1. $X \in \mathcal{A}$
2. $\emptyset \in \mathcal{A}$
3. $\forall A_i \in \mathcal{A}, \cap_{i=1}^{n} A_i \in \mathcal{A}$
4. $\forall A_i \in \mathcal{A}, \cup_{i=1}^{n} A_i \in \mathcal{A}$
5. $\forall A_i \in \mathcal{A}, A_i^c \in \mathcal{A}$

Example 1.15. Given a set $X$, the set of parts of $X$ is a $\sigma$-algebra which receives a special name of discrete $\sigma$-algebra.

Example 1.16. A special $\sigma$-algebra is the Borel $\sigma$-algebra, usually called just Borel algebra. Defined as $\sigma$-algebras that can be build by union, intersections and complements between the open sets of a topology.

Definition 1.17. An Exterior Measure in a set $X$ is a function $\mu : P(X) \rightarrow \mathbb{R}^+$ such that

1. $\mu(\emptyset) = 0$
2. $\forall A, B \subset X$, if $A \subseteq B$, then $\mu(A) \leq \mu(B)$
3. $\forall A_i \subset X$, $\mu(\cup_{i=1}^{\infty} A_i) \leq \sum_{i=1}^{\infty} \mu(A_i)$

Example 1.18. A intuitively known measure is the Lebesgue Measure. It corresponds to the basic notion of volume in $\mathbb{R}^n$.

Example 1.19. The Gaussian Measure in the line of a subset $A \subset \mathbb{R}$ is defined as

$$\gamma(A) = \frac{1}{\sqrt{2\pi}} \int_{A} e^{-\frac{1}{2}x^2} dx$$

such that $x \in \mathbb{R}$. 

4
**Definition 1.20.** A **Probability Space** is a triple $P = (\Omega, \mathcal{A}, \mu)$, such that $\Omega$ is a set, $\mathcal{A}$ is a $\sigma$-algebra in $\Omega$ and $\mu$ is an exterior measure in $\mathcal{A}$, in a way that $\mu(\Omega) = 1$.

We call $\Omega$ the **Sample Space**;
$\mathcal{A}$ is the **event set**;
$\mu$ is a **probability function** and
$\mu(A) \mid A \in \mathcal{A}$ is the **probability** of the happening of the event $A$.

**Example 1.21.** A dice roll can be described by the set $\Omega = \{1, 2, 3, 4, 5, 6\}$. The probability of each value in this set be the value of the roll can be given by a function $\mu$. The probability of an odd number be the result is given by $\mu(\{1, 3, 5\})$. If we have a FAIR dice, it means the results are equiprobable, we have $\mu(\{1, 3, 5\}) = 1/2$. In this case, an interesting $\sigma$-algebra would be the discrete $\sigma$-algebra.

### 1.3 useful Graph Theory Results

The graph theory came up with Euler and the Königsberg’s bridge problem. We can think of graphs like a set of points connected by line segments. Each point represents an object which we call vertex. Each line represents a relation between vertices and we call the lines edges. Interesting examples of graphs can be cities connected by roads, users in a social web that are connected by a friendly relationship in the program. The edges can be directed, it would mean an edge is an unilateral relation. A graph with directed edges is a directed graph.

**Definition 1.22.** Let $G(V, A)$ be a graph, in which $V = \{v_1, v_2, ...\}$ is the set of vertex and $A = \{a_1, a_2, ...\}$ the set of edges. We say $G$ is a **simple graph** if, and only if, two different edges connect different vertices and there isn’t any edge that connect a vertex in itself.

**Definition 1.23.** A simple graph is called a **complete graph** if, and only if, every pair of vertex is connected by one edge.

We can be interested, for example, in find the number of edges a complete graph with $n$ vertex. That is what we are going to do.

**Theorem 1.24.** Let $G(V, A) = K_n$ be a complete graph with $n$ vertices. The number of edges of the graph is $\frac{n(n-1)}{2}$.

**Proof.** We can go through a combinatory approach. We know every edge connect exactly two vertices. This way, we can simple use a combination to calculate the number of possible edges for a given number of vertices $n$.

We use $C_n^2 = \frac{n!}{(n-2)!2!} = \frac{n(n-1)(n-2)!}{(n-2)!2} = \frac{n(n-1)}{2}$ then we finish the demonstration. \qed
2 Automata in Topological Spaces

Definition 2.1 (Automaton). An automaton is a triple \( A = (Q, \Gamma, \delta) \), such that

- \( Q \) is the set of states of the automaton;
- \( \Gamma \) is a set called the input alphabet or input set of the automaton and \( \Gamma^* \) is the set of strings of \( \Gamma \), built by the Kleene star over \( \Gamma \);
- \( \delta : Q \times \Gamma^* \to Q \) is called the transition function, satisfying \( \delta(q, e) = q \) for all \( q \in Q \) and a given \( e \in \Gamma^* \) named empty string and \( \delta(q, ab) = \delta(\delta(q, a), b) \) for all \( a, b \in \Gamma^* \).

\( \Gamma \) is not necessarily a set of words as the ones used in human language, is a convention to use this term to refer to anything that can be "read" or understood by an automaton, being it sensor signs, mechanical stimuli, or even human words. This way we can handle concepts like "letter", "alphabet" and "word" in a broad sense.

As pointed in [6], there is at least one topology that for \( Q \) that makes it a topological space. Hence, we can associate to the set of states of the automaton, topological properties. Since \( Q \) is directly related with the automaton it defines, it is possible to associate topological properties to the automaton itself, reflecting the features of its set of states.

Definition 2.2. Suppose an automaton \( A = (Q, \Gamma, \delta) \) and define the source operator \( \sigma \) in a way that for a subset \( B \subset Q \), \( \sigma(B) = \{ q \in Q \mid \delta(q, m) \in B, m \in \Gamma^* \} \). The set \( \sigma(B) \) is called source of \( B \).

As showed in [2] and reinforced in [6], \( \sigma \) is a closure operator in \( Q \) and, then, defines a topology for \( Q \), which we call source topology, \( T_\sigma \). The first thing we can notice is the source topology is related with the transition structure of the automaton, so we can predict the existence of relations between properties in automata theory and topological properties of \( Q \).

Some of the theorems presented in this article were proved in [3], [1], [6] and [5]. In addition, in these papers we can find other interesting results. We present some of them here in order to make easier to the reader access them, so it will not be necessary to search for another paper to see the basics properties and theorems in the area.

Definition 2.3. Given an automaton \( A = (Q, \Gamma, \delta) \), we define a subautomaton \( A' \) of \( A \) as \( A' = (Q', \Gamma, \delta') \), such that \( Q' \subset Q \) and \( \delta' \) is the restriction of \( \delta \) in \( Q' \).

Definition 2.4. Let \( A = (Q, \Gamma, \delta) \) be an automaton and \( A' = (Q', \Gamma, \delta') \) a subautomaton of \( A \). Considering \( \{q_1, q_2, \ldots\} = (Q - Q') \neq \emptyset \) and \( \{\gamma_1, \gamma_2, \ldots\} = \Gamma^* \). \( A' \) is said separated in \( A \) if, and only if, \( \{\delta(q_i, \gamma_j) \mid 1 \leq i, 1 \leq j \} \cap Q' = \emptyset \). \( A \) is said connected if it’s not possible to define a separated subautomaton in \( A \).

Definition 2.5. An automaton is retrievable if, and only if, for all \( p, q \in Q \), and any \( m \in \Gamma^* \) such that \( \delta(q, m) = p \), there is some \( x \in \Gamma^* \) in a way that \( \delta(p, x) = q \).
We can say, intuitively, that any transition can be undone, existing, then, some string that can produce this result. We must pay attention in the fact that it doesn’t has to exist only one string that can undone some transition or all transitions but, in general, for every transition, there is a string that undone it, these strings can be different for every transition.

**Definition 2.6.** An automaton is called **reversible** if, and only if, for all \( p, q \in Q \) and some \( m \) such that \( \delta(q, m) = p \), there is some \( x \in \Gamma \) in a way that \( \delta(p, x) = q \).

Notice the difference between retrievability and reversibility. While retrievable automata needs a string to undone some transition, a reversible automata only needs a letter. We can see every reversible automaton is also retrievable.

**Theorem 2.7.** If, and only if, an automaton is connected, then it is also retrievable.

**Proof.** (Retrievable) ⇒ (Connected): 
If an automaton \( A = (Q, \Gamma, \delta) \) is not connected, so we can find a separated subautomaton \( A' = (Q', \Gamma, \delta') \). By separated subautomaton’s definition: Let \( Q_s \subset Q \) a non empty set in a way \( Q_s = (Q - Q') = \{q_1, q_2, ..., \}, \{\delta(q, x) \mid \forall x \in \Gamma^*, 1 \leq i \} \cap Q' = \emptyset \). So there is no transition from the set \( Q_s \) to the set \( Q' \), but it must exist some transition from the set \( Q' \) to the set \( Q_s \), otherwise, one of the sets would not be part of the automaton, because it could not be reached.

So must exist at least one irreversible transition, it means there is no way to undone this transition, characterizing a non retrievable automaton. We have showed that if an automaton is not connected, then it is not retrievable, taking the negative of the consequence we conclude that if an automaton is retrievable, then it is connected.

(Connected) ⇒ (Retrievable):
If an automaton is not retrievable, then must exist some transition which can not be undone, what allows us to find a separated subautomaton. It is enough if the subautomaton have one of the states of the transition and doesn’t have the other. Specifically, if the irreversible transition is \( \delta(q', m) = q \), a subautomaton with the set of states equal the source of \( \{q'\} \) will be separated.

In this case, we have showed that if an automaton is not retrievable, then it is not connected. So we conclude that if the automaton is connected, then it is retrievable.

**Definition 2.8.** An automaton is called **weakly connected** if, and only if, for all \( p, q \in Q \), there is at least one \( m \in \Gamma^* \) such that \( \delta(q, m) = p \).

**Definition 2.9.** An automaton is said to be **strongly connected** if, and only if, for all \( p, q \in Q \), there is at least one \( x \in \Gamma \) such that \( \delta(q, x) = p \).

**Theorem 2.10.** If an automaton is weakly connected, then it is also retrievable.

**Proof.** Let \( A = (Q, \Gamma, \delta) \) be a weakly connected automaton. Suppose any pair of states \( p, q \) and the transition \( \delta(p, m) = q \), for some \( m \in \Gamma^* \). If the automaton
is not retrievable, then can’t exist some string $m^{-1}$ such that $\delta(q, m^{-1}) = p$, which is an absurd, because if there wasn’t a string that allows a transition from $p$ to $q$, the automaton wouldn’t be weakly connected.

**Theorem 2.11.** If an automaton is strongly connected, then it is also reversible.

**Proof.** The proof is analogous to the last theorem. Just changing the strings for letters would be enough.

**Definition 2.12.** Let $A = (Q, \Gamma, \delta)$ be an automaton. Select $p, q \in Q$ and $m \in \Gamma^*$. We call $q$ a $m$-successor of $p$ and we say $p$ is a $m$-predecessor of $q$ if, and only if, $\delta(p, m) = q$.

We can see the source topology is closely related with the notion of successor and predecessor.

**Definition 2.13.** We say an automaton $A = (Q, \Gamma, \delta)$ has unique predecessors if, and only if, for all $q, p \in Q, m \in \Gamma$, $\delta(q, m) = \delta(p, m) \Rightarrow p = q$.

Organizing the properties here presented and considering the relations between them and its implications, we graph this diagram:

$$
\begin{array}{c}
P_1 \\
\bigg\uparrow & \downarrow \bigg \downarrow \\
\bigg \downarrow & \quad \quad \quad \\
\bigg \downarrow & \\
P_2 & \\
\bigg \uparrow & \downarrow \bigg \downarrow \\
\bigg \downarrow & \\
P_3 & \\
\bigg \uparrow & \downarrow \bigg \downarrow \\
\bigg \downarrow & \quad \quad \quad \\
\bigg \downarrow & \\
P_4 & \\
\quad \quad \quad \\
\bigg \uparrow & \downarrow \bigg \downarrow \\
\bigg \downarrow & \\
P_5 & \\
\end{array}
$$

The properties are presented as the Table 1. Arrows represents implications. For example, the property 1 (Strongly connected) implies property 2 (Reversibility), which also implies 4 (Retrievability). 4 and 5 implies each other.

| $P_1$ | Strongly Connected |
|-------|-------------------|
| $P_2$ | Reversible        |
| $P_3$ | Weakly Connected  |
| $P_4$ | Retrievable       |
| $P_5$ | Connected         |

**Theorem 2.14.** The following properties are inherited from automata by its subautomata:

i Strongly connectivity

ii Reversibility

8
ii Analogously, if two states of the automaton are in the subautomaton, the reversibility feature is inherited by the same property of the transition function. Both the transition and its inverse are maintained.

iii Analyzing the automaton $A = (Q, \Gamma, \delta)$ and a subautomaton $A' = (Q', \Gamma, \delta')$ of $A$, since $Q' \subset Q$, there will not be subset of $Q'$ which is not a subset of $Q$, it means, any subautomaton of $A'$ is also a subautomaton of $A$. If $A$ is connected, $A'$ must be connected too, since the existence of a separated subautomaton in $A'$ would imply the existence of a separated in $A$, which can not be the case since it is connected.

Theorem 2.15. Let $A = (Q, \Gamma, \delta)$ be an automaton. If, and only if, the source topology $\mathcal{T}_s$ is indiscrete, then $A$ is connected.

Proof. $(\text{connected}) \Rightarrow (\text{Indiscrete Topology})$

If $A$ is connected, it means the source of any subset of $Q$ is itself, because all the states allow transition to every other state. It is, for every subset $B \subset Q$, $\sigma(B) = Q$ by definition of connected automaton. This way, the unique open sets will be $\emptyset = \sigma(Q)^C$ and $Q = \sigma(\emptyset)^C$ since $\sigma(\emptyset) = \emptyset$.

$(\text{Indiscrete Topology}) \Rightarrow (\text{Weakly Connected})$

If the topology is indiscrete, the source of every subset of $Q$ must be $Q$ or $\emptyset$. Since $Q' \subseteq \sigma(Q') \forall Q' \in Q$, a non-empty subset of $Q$ will have source $Q$, which is not possible if there were a separated subautomaton.

Theorem 2.16. Let $A' = (Q', \Gamma, \delta')$ be a subautomaton of $A = (Q, \Gamma, \delta)$. $A'$ is separated if, and only if, $\sigma(Q') = Q'$.

Proof. $(\text{Separated}) \Rightarrow (\sigma(Q') = Q')$

If $A'$ is separated, there won’t be any state $q \in Q, q \notin Q'$ such that $\delta(q, m) = q'$ for some $m \in \Gamma^*$ and some $q' \in Q'$. This way, it is impossible that $\sigma(Q')$ has more elements than $Q'$. But since $C \subseteq \sigma(C)$ for any subset $C$, we conclude that $\sigma(Q') = Q'$.

$(\sigma(Q') = Q') \Rightarrow (\text{Separated})$

We consider now the case in which $A'$ is not separated. If $A'$ is not separated, there exist $q \in Q, q \notin Q'$ such that $\delta(q, m) = q'$ for some $m \in \Gamma^*$ and some $q' \in Q'$. It means there is predecessor for at least one $q' \in Q'$. Then, by definition of source operator, $\sigma(Q') \neq (Q')$. Since $A'$ being not separated implies $\sigma(Q') \neq Q'$. The converse shows us that if $\sigma(Q') = Q$, then $A'$ is separated.
BAVEL shows us that an automaton is retrievable only if \( \sigma(Q') = Q' \), \( \forall Q' \subset Q \), which contradicts other two theorems that Bavel himself proves. He argues that an automaton is retrievable only if it is a disjoint union of weakly connected automata and in a weakly connected automaton holds that \( \sigma(Q') = Q \). Of course it needs to exist some transition between the disjoint automata, otherwise, it wouldn’t make sense to say they are the same automaton. If there is one or more transitions from one subautomaton to another one (considering they are disjoint automata) and there isn’t a way to undo it, the automaton is not retrievable. If there is a way to undone the transitions between disjoint weakly connected subautomata, it will be, then, a retrievable automaton, since the retrievability of the subautomata are guaranteed by theorem 2.10. But in this case, since the disjoint subautomata are weakly connected and there are transitions in "both hands" between them, any subset will have source equal the whole set of states of the automaton, which disagree with BAVEL's theorem. These results used in our arguing are also proved by BAVEL and we showed that they contradict the statement that \( \sigma(Q') = Q' \) holds for a retrievable automaton.

We must just be careful at the definitions used, since our concept of weakly connected automaton corresponds to strongly connected automaton in BAVEL’s work.

The next theorem is a non-conflicting version of the statement discussed above. We want to reformulate the theorem and prove it.

**Theorem 2.17.** An automaton \( A = (Q, \Gamma, \delta) \) is retrievable if, and only if, \( \sigma(Q') \neq Q' \) \( \forall Q' \subset Q \).

**Proof.** (Retrievable) \( \Rightarrow \) (\( \sigma(Q') \neq Q' \))

If the automaton \( A \) is retrievable, then it is connected by theorem 2.7, so \( A \) can’t have a separated subautomaton. By theorem 2.16, a separated subautomaton \( A' = (Q', \Gamma, \delta) \) has \( \sigma(Q') = Q' \). Since \( A \) can’t have separated subautomaton, can’t exist \( Q' \subset Q \) such that \( \sigma(Q') = Q' \).

(\( \sigma(Q') \neq Q' \)) \( \Rightarrow \) (Recuperável)

If a subautomaton has \( \sigma(Q') \neq Q' \), it can’t be separated since theorem 2.16. If \( \sigma(Q') \neq Q' \) \( \forall Q' \subset Q \), the automaton is connected, because there won’t be separated subautomata. This implies, by theorem 2.7, that it is a retrievable automaton. \( \square \)

**Theorem 2.18.** There is no automaton with more than one state and source topology equal the discrete topology.

**Proof.** Suppose an automaton \( A = (Q, \Gamma, \delta) \) with more than one state. In order to the source topology be discrete, the set \( Q - \{q\} \), for an element \( q \in Q \), must be an open set. Hence, it must exist some \( q' \in Q \) such that \( \sigma(\{q'\}) = \{q\} \). Since the source operator has the following property: \( R \subseteq \sigma(R) \) \( \forall R \subset Q \), the only possibility is that \( q' = q \). In addition, our supposition doesn’t allows the existence of some state that have transition to \( q \), because it would make \( \sigma(\{q'\}) \neq \{q\} \), \( q \) needs to have a transition to some state, or it will be the only
state of the automaton. We conclude there is a \( p \) such that \( \delta(q,m) = p \) for some \( m \in \Gamma^* \). Again, since we want a discrete topology, \( Q - \{ p \} \) must be a open set. In order to this happening, it must exist some \( r \in Q \) in a way that \( \sigma(\{r\}) = \{p\} \), what is impossible if \( r \neq p \), because \( \sigma(\{r\}) \) contains \( r \). It would give us \( \{r,p\} \subseteq \sigma(\{r\}) \).

It left us with the last possibility of \( r = p \). Even this way we would have a problem, because there is a transition from \( q \) to \( p \), so \( q \in \sigma(\{p\}) \). Then, we will never have an open set \( Q - \{ p \} \) and, hence, the topology will never be discrete if the automaton have more than one state.

\[
\square
\]

**Definition 2.19 (Equality Between Automata).** Let \( A = (Q_1, \Gamma_1, \delta_1) \) and \( B = (Q_2, \Gamma_2, \delta_2) \) be two automata, we say \( A \) is equal \( B \) if, and only if,

1. \( Q_1 \approx Q_2 \), being \( f : Q_1 \to Q_2 \) a homeomorphism;
2. \( \Gamma_1 = \Gamma_2 \);
3. \( f(\text{Im}(\delta_1)) = \text{Im}(\delta_2) \) and \( f^*(\text{Im}(\delta_2)) = \text{Im}(\delta_1) \), it is, \( \delta_1 \sim \delta_2 \).

Equal automata share the same properties and are, essentially, the same automaton.

**Definition 2.20 (Homeomorphism of Automata).** Two automata are homeomorphic if there is a homeomorphism between its set of states.

Homeomorphic automata share all the topological invariant properties.

We want to create one more correlation between automata. One that allows us to identify when two automata have the same properties even if there isn’t a clear similarity between them. So lets consider the transition function \( \delta \). Notice that it is actually a two variable function, but lets decompose it in many one-variable functions.

For each \( m \in \Gamma^* \) there is an unique function \( \delta_m : Q \to Q \), such that \( \delta_m(q) = \delta(q,m) \). This way we will have a set of functions with same cardinality as \( \Gamma^* \). Now we have a set of endomorphisms that does the same as the transition function.

We can, then, create a second definition of an automaton.

**Definition 2.21 (Alternative).** An automaton \( A = (Q, \Gamma, \delta) \) can also be represented by \( A = (Q, \delta_1, \delta_2, ...) \) such that \( \delta_i(q) = \delta(q,m_i) \) with \( m_i \in \Gamma^* \).

**Definition 2.22 (Congruent automata).** Let \( A = (Q, \delta_1, \delta_2, ...) \) and \( A' = (Q, \delta'_1, \delta'_2, ...) \) be two automata. Consider the sets \( G = \{\delta_1, \delta_2, ...\} \) and \( G' = \{\delta'_1, \delta'_2, ...\} \).

We say \( A' \) is congruent to \( A \) if there is a homeomorphism \( f : G' \to G \) that satisfies: If \( f(\delta'_i) = \delta_i \) and \( f(\delta'_j) = \delta_j \), so \( f(\delta'_i \circ \delta'_j) = \delta_i \circ \delta_j \), such that \( \circ \) represents a composition of functions.

**Theorem 2.23.** The following properties are inherited by automata congruence:
i retrievability;

ii connectivity

iii reversibility;

iv weak connectivity;

v strong connectivity.

Proof. i If there is some $x \in \Gamma^*$ which reverts a transition $\delta_\gamma(q)$ for any $\gamma \in \Gamma^*$, it means there is a function $\delta_x | \delta_x(\delta_\gamma(q)) = q$.

By the definition of congruence of automata, there is a $f$ such that $f(\delta_x) = \delta'_x$ and $f(\delta_\gamma) = \delta'_\gamma$, with the property $f(\delta_x(\delta_\gamma(q))) = \delta'_x(\delta'_\gamma(q'))$, which shows that the automaton $A'$ preserves retrievability in congruence.

ii Since there is the relation connectivity $\Leftrightarrow$ retrievability, an automaton which is connected is also retrievable. So any congruent automaton will be also retrievable since the proof above and then, will be connected.

iii The proof is analogous to the retrievability version, but we use $\gamma, x \in \Gamma$ instead of $x \in \Gamma^*$.

iv A weakly connected automaton allows transition from any state with some string $m \in \Gamma^*$. But given a transition $\delta_m(q) = p$, exists, by the homeomorphism, $\delta'_m(q') = p'$. Using the property of the homeomorphism based on the definition of congruence of automaton, any other transition from $p$ to some $t$, using $\delta_n(p) = t | n \in \Gamma^*$, can be built with

$f(\delta_m(\delta_n(t))) = \delta'_m(\delta'_n(t')) = t'$. Which is valid for any $t$, including $t = q$, it means, any transitions that can be undone by some transition, will have its equivalent in the congruent automaton, as well as its -reverse transition.

v The proof is analogous, but, we assume $n, m \in \Gamma$.

3 Finite Automata

The discussed concepts are applicable to any automaton. But most of the automata in practice are Finite Automata. Finite automata are the ones which have a finite set of states. Given the importance of this kind of automata, we are going to dedicate a whole section to talk about the modeling of finite automata.

We want to define the two main kinds of finite automata, the deterministic and the probabilistic. We are also going to give a special attention to the probabilistic ones.
3.1 Deterministic Finite Automata

From now, we will define and present a property of deterministic finite automata. Deterministic automata are very well known in academy.

**Definition 3.1.** A **Deterministic Finite Automaton** is a 5-tuple $A_{FD} = (Q, \Gamma, \delta, q_0, F)$ in which
- $Q$ is a finite set, the set of states of the automaton;
- $\Gamma$ is the input alphabet;
- $\delta : Q \times \Gamma^* \rightarrow Q$ is the transition function, which satisfies $\delta(q,e) = q$ for all $q \in Q$ and some $e \in \Gamma^*$, the empty string, with the addition of $\delta(q,ab) = \delta(\delta(q,a),b)$ for all $a,b \in \Gamma^*$ and all $q \in Q$;
- $q_0 \in Q$ is the initial set of the automaton;
- $F \subset Q$ it is the set of final states and when the automaton is on some state $q_f \in F$, it stops its work.

With the feature that, for all $q, p \in Q, m \in \Gamma^*$, if $\delta(p,m) = \delta(q,m)$, then $p = q$.

This kind of automaton is said to be deterministic exactly because of its last feature: for some transition given by a pair of string and state, there is an unique resultant state.

We could think about representing an automaton as a directed graph. This, in fact, is often used in state-machine analyzes. The states of an automaton are presented as vertices of the graph, while the transitions are the edges.

A complete graph, in this approach, would mean a strongly connected automaton. The difference is in the fact we need to think about a complete graph and consider the two ways possible for directing the transitions (edges).

**Theorem 3.2.** A strongly connected deterministic finite automaton with $n$ states has at least $n(n-1)$ transitions.

*Proof.* Using the result given presented in 1.24, we consider the transitions as edges and conclude that the automaton must have $\frac{n(n-1)}{2}$ edges. But we also need to care about the direction of the edges and there are two directions possible for each edge. So we can just multiply the number of edges by 2 and we are counting, then, both directions. We finish with $n(n - 1)$ edges.

The number of transitions could also be more because there would be more than one transition from a state $q$ to a state $p$ using different strings. Then, the number we got is the minimum number of edges, or transitions.

It is possible to find an existence condition for a finite strongly connected automaton has unique predecessors. Writing the number of elements of the set of states as $n(Q)$ and the number of letters in the alphabet as $n(\Gamma)$, we can notice that $n(Q) - 1 \leq n(\Gamma)$, so there won’t be distinct states allowing transition to the same state using the same letter.

If the automaton has $n(Q)$ states and it is strongly connected, then must be, for each state, $n(Q) - 1$ transitions. Supposing a generic state $q$, each transition can be represented by an arrow going from a state to $q$. It must exist at least
3.2 Probabilistic Finite Automata

Intuitively, the difference between deterministic and probabilistic automata is that, while, for a deterministic automaton, the transition function of a given pair state-string can lead us to an unique new state, for a probabilistic automaton, it could give us more than just one new state.

In an algebraic approach, we are going to define probabilistic automata using probability spaces.

**Definition 3.3.** A Probabilistic Finite Automaton is a 5-tuple \( A_{FP} = (Q, \Gamma, \delta_p, q_0, F) \) in which
- \( Q \) is a finite set of states;
- \( \Gamma \) is the input alphabet;
- \( \delta : Q \times \Gamma^* \rightarrow \mathcal{P}(Q) \) is the transition function, in which \( \mathcal{P}(Q) \) means the power set of \( Q \);
- \( q_0 \in Q \) is the initial state of the automaton;
- \( F \subseteq Q \) is the set of final states of the automaton.

This is the standard definition of a probabilistic finite automaton (PFA). Some articles discuss the interpretation of a probabilistic automaton. We are going to assume that the transition is made by the verification of a random variable and using its value to decide the next state. Based on this idea, we can bring a new definition using probability spaces.

**Definition 3.4 (Alternative).** A Probabilistic Finite Automaton is a 5-tuple \( A_{FP} = (P, \Gamma, \delta_p, q_0, F) \) in which
- \( P = \{P_{m_1,q_1}, P_{m_1,q_2}, \ldots, P_{m_1,q_n}, P_{m_2,q_1}, \ldots\} \) is a class of probability spaces;
- \( P_{m_i,q_j} = (Q, \mathcal{A}, \mu_{m_i,q_j}) \);
- \( \mu_{m_i,q_j} : Q \rightarrow [0,1] \) is a function that receives a state \( q \) as input and gives the probability of the transition from \( q_j \) to \( q \) by a string \( m_i \);
- \( Q \) is the set of states of the automaton and \( \mathcal{A} \) is a \( \sigma \)-algebra in \( Q \);
- \( \Gamma \) is the input alphabet of the automaton;
- \( \delta_p : Q \times \Gamma^* \rightarrow (Q \times [0,1])^n \) in which \( n \) is the number of elements of \( Q \), in a way that \( \delta_p(q_i, m_j) = \{(q_1, \mu_{m_j,q_1}(\{q_1\})), (q_2, \mu_{m_j,q_1}(\{q_2\})), \ldots, (q_n, \mu_{m_j,q_1}(\{q_n\}))\} \);
- \( q_0 \in Q \) is the initial state of the automaton;
- \( F \subseteq Q \) is the set of final states of the automaton.

The first observations we can do is that the set of states now is a sample space, subsets of \( Q \) are events and for each pair string-state, there is a probability function. The transition function \( \delta_p \) just gives us pairs of a state and its respective transition probability. When dealing with a subset with more than one state, we are computing the probability that the transition lead to any state inside this subset.

Another thing that can be noticed came from the definition of probability measure. Considering each state as a subset of \( Q \), the union of all the states
gives \( Q \) as result. By one of the conditions of the definition of exterior measure, we can see that 
\[
\mu_{m_{i_{q_{i}}}}(\bigcup_{j=1}^{m} \{ q_j \}) \leq \sum_{p=1}^{n} \mu_{m_{i_{q_{j}}}}(\{ q_p \}).
\]
But since the union of all states is \( Q \) itself, we get 
\[
\mu_{m_{i_{q_{i}}}}(Q) \leq \sum_{p=1}^{n} \mu_{m_{i_{q_{j}}}}(\{ q_p \}).
\]
We also must remember that \( \mu(\Omega) = 1 \). The probability of an event can't be greater than 1, we conclude that the sum of the probabilities of all transitions is 1.

But notice we are not summing the probability related to subsets involving more than one state, like \( \mu_{m_{i_{q_{i}}}}(\{ q_1, q_2 \}) \), but only the isolated probabilities of the transition to each state.

With the classical definition of automaton, we saw that \( \delta(q, nm) = \delta(q, n), m \). An interesting question that could appear is how to extend this notion to our alternative definition of probabilistic automaton.

By the multiplicative principle, we just need to multiply the probability related to the first string with the probability related with the second string. We have 
\[
\delta_p(q_1, m_1m_2) = \{ (q_i, \sum_{k=1}^{n} \mu_{m_{i_{q_{k}}}}(\{ q_k \}) \mu_{m_{i_{q_{l}}}}(\{ q_l \})) \mid 1 \leq l \leq n \} \text{ in which } n \text{ is the number of states of the automaton. It gives us the probability of each state be reached from another given state and a chain of two strings.}
\]

The summation is just the sum of the probabilities of all the possible transitions that starts in a state \( q \) and finishes in a state \( q_l \) by making the whole transition in two steps, two smaller transitions (one for each string).

We can generalize this formula for a chain \( x \) of strings: 
\[
\delta_p(q_1, m_1m_2...m_x) = \{ (q_i, \sum_{k_1=1}^{n_{i_{q_{i}}}} \sum_{k_2=1}^{n_{i_{q_{j}}}} \ldots \sum_{k_{x-1}=1}^{n_{i_{q_{l}}}} \mu_{m_{i_{q_{k_{l}}}}}((\{ q_{l_{i}} \}) \prod_{j=1}^{x-1} \mu_{m_{j_{q_{a}}}}((\{ q_{a_{j}} \})) \mid 1 \leq l \leq n \} \text{. Again, } n \text{ is the number of states of the automaton.}
\]

**Theorem 3.5.** Given a chain of \( x \) strings, generated randomly in an alphabet of \( p \) strings, the chance of this chain leads successfully from a state \( q_1 \) to a state \( q_f \) is

\[
\mathcal{P}(q_1 \rightarrow q_f) = \sum_{i=1}^{p^x} \sum_{k_{x-1}=1}^{n_{i_{q_{l}}}} \mu_{m_{i_{q_{k_{l}}}}}(\{ q_{k_{l}} \}) \prod_{j=1}^{x-1} \mu_{m_{j_{q_{a}}}}((\{ q_{a_{j}} \}))
\]

(1)

Where \( P_{m_{i_{q_{k}}}}(q_{k_{l}}) \) is the probability of the i-th chain of strings possible be chosen between all the possible chains, with \( m_{i_{q_{k}}} \in \Gamma^* \forall j \).

\( p^x \) is the number of possible chains with \( x \) strings, resulting from a finite alphabet of \( p \) strings. In the case of infinity strings, one could just introduce the limit \( p^x \rightarrow \infty \).

**Proof.** Calling \( P_{q_{i,A}}(q_f) \) the probability of the transition from the state \( q_1 \) to \( q_f \) along the chain \( A \). The probability of a given chain \( A \) with \( x \) strings in this alphabet be chosen in a random process between all the chains possible is \( P_A \).

By the multiplicative principle, the probability of choosing this chain \( A \) and it successfully leads from \( q \) to \( q_f \) is \( P_A P_{q_{i,A}}(q_f) \).

If we sum this probability for all the possible chains, we will have the probability of a random chain makes the interested transition. This can be written as 
\[
\sum_{i=1}^{p^x} P_A P_{q_{i,A}}(q_f),
\]

having \( A_i \) as the i-th chain possible and \( p^x \) as the number of chains possible. The term \( P_{q_{i,A}}(q_f) \) is exactly the f-th element of the set.
\[ \delta_p(q, A) \] and, then, we can replace the relation already showed to the probability of a given transition by a certain chain of strings, getting

\[
\sum_{i=1}^{p^*} P_{m_i \ldots m_i^n} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \ldots \sum_{k_{x-1}=1}^{n} \mu_{m_i q_k \ldots q_k} ((q_f)) \prod_{j=1}^{x-1} \mu_{m_j q_{k_{x-j}}} ((q_{k_j}))
\]

, which is the probability of, choosing a random chain of \( x \) strings and giving it as input to the automaton, it results in the transition from \( q_1 \) to \( q_f \). Notice that \( m_i^j \) is the \( j \)-th string of the \( i \)-th chain possible \( A_i \).

We can even think about a equiprobable random process, in which every chain has the same chance to be chosen, \( \frac{1}{p^*} \). We get

\[
\frac{1}{p^*} \sum_{i=1}^{p^*} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \ldots \sum_{k_{x-1}=1}^{n} \mu_{m_i q_k \ldots q_k} ((q_f)) \prod_{j=1}^{x-1} \mu_{m_j q_{k_{x-j}}} ((q_{k_j})).
\]

This last situation would only be possible in a automaton with a finite input alphabet, since the probability of each chain would go to zero if the number of strings increases to infinity.

A last note in probabilistic automata is that every deterministic automaton is also probabilistic. It is true since we can build a Deterministic Finite Automaton (DFA) using all the ideas from a PFA. Indeed, let \( p, q \) be two states of a DFA such that \( \delta(q, m) = p \), for some \( m \in \Gamma^* \). We could represent this transition in a PFA in which \( \delta(q, m) \) gives us a probability 1 to the state \( p \) and a probability 0 for any other state different of \( p \). This can be made for any transition of a DFA, so a DFA is just a PFA which transitions have probability 1 or 0.

4 Automata Association

Eventually, automata are used in groups. Two or more automata works simultaneously. Many times we can be interested in the combined results of the automata, analyzing a "global" situation instead of looking for each single automaton at a time.

The problem can became very complex when there are interactions between the automata, it is, the states and transitions of one interfere in the transitions or states of the other one. This kind of relation is very common in many systems, like security systems, where some actions are not allowed if some condition is not satisfied.

Lets model what we will call Dependence Forms Between Automata. Our goal is to formalize dependencies and relations between automata that works together. We want to build patterns for a "final automaton" which is equivalent to the system with many automata working in a group.

We are going to be able to build a resultant automaton, an unique automaton that contains all the features about the automata in the system and its different relations.

4.1 Independent Automata

We are going to call Independent automata, two or more automata that can be described by an unique automaton and the transitions, states and alpha-
Theorem 4.4. The Resultant Automaton of a system of many automata is a single automaton that describes the whole system.

Definition 4.2. Let \( A = (Q_A, \Gamma_A, \delta_A) \) and \( B = (Q_B, \Gamma_B, \delta_B) \) be two deterministic automata. We call \( A \) and \( B \) independent automata if, and only if, the resultant automaton \( A \otimes B \) is given by \( A \otimes B = (Q_A \times Q_B, \Gamma_A \times \Gamma_B, \delta_{AB}) \), with \( \delta_{AB}(q_1, m) = \delta_{AB}((q_A, q_B), (m_A, m_B)) = q_2 = (\delta_A(q_A, m_A), \delta_B(q_B, m_B)) \forall q_A \in Q_A, q_B \in Q_B, m_A \in \Gamma_A^*, m_B \in \Gamma_B^* \).

We can write the resultant automaton in a reduced form by \( R = (Q_R, \Gamma_R, \delta_R) \). Notice the transition function can be represented by \( \delta_R : (Q_A \times Q_B) \times (\Gamma_A \times \Gamma_B) \to (Q_A \times Q_B) \), as well as \( \delta_R : Q_R \times \Gamma_R \to Q_R \).

Definition 4.3. Let \( A = (P_A, \Gamma_A, \delta_A, q_A) \) and \( B = (P_B, \Gamma_B, \delta_B, q_B, F_B) \) two probabilistic automata. We call \( A \) and \( B \) independent automata if, and only if, the resultant automaton is given by \( A \otimes B = (P_{AB}, \Gamma_A \times \Gamma_B, \delta, q_0, F) \).

In which \( P_{AB} = \{P_{m_1,q_1}, P_{m_2,q_2}, ..., P_{m_i,q_i}, ...\} \). The probability spaces are \( P_{m_i,q_i} = (Q_A \times Q_B, \mathcal{F}, \mu_{m_i,q_i}) \), with \( m_i \in (\Gamma_A \times \Gamma_B)^* \) and \( q_i \in Q_A \times Q_B \) is in a way that \( m_i = (m_i^A, m_i^B) \) and \( q_i = (q_i^A, q_i^B) \), so \( \mu_{m_i,q_i} = \mu_{m_i^A,q_i^A} \mu_{m_i^B,q_i^B} \). Which is consequence of the multiplicative principle.

This definition is more general since every deterministic automaton is also probabilistic.

Theorem 4.4. If two automata \( A \) and \( B \) are independent, the source topology of the resultant automaton is the product topology of the source topology of \( A \) and \( B \).

Proof. The product topology is defined for \( Q_A \times Q_B \) and it is a natural topology for its space. But there is also the source topology of the resultant automaton. What we need to show is that the two topologies are actually the same.

Suppose two open sets of the source topology \( U_A \in \mathcal{T}_A \) and \( U_B \in \mathcal{T}_B \). \( U_A \times U_B \) is an open set of the product topology \( \mathcal{T}_{AB} \). \( V_A \) and \( V_B \), such that \( \sigma_A(V_A)^C = U_A \) and \( \sigma_B(V_B)^C = U_B \). \( \sigma_{AB}(V_A \times V_B) = \{q_0 \mid \delta_{AB}(q_0, m) \in V_A \times V_B, \forall m \in (\Gamma_A \times \Gamma_B)^*\} \). Since \( \delta_{AB}(q, x) = (\delta_A(q_A, x_A), \delta_B(q_B, x_B)) \), we can verify that \( \sigma_{AB}(V_A \times V_B) = \{q_{A0}, q_{B0} \mid \delta_A(q_{A0}, m_A) \in V_A \text{ and } \delta_B(q_{B0}, m_B) \in V_B, \forall m_A \in \Gamma_A, \text{ and } m_B \in \Gamma_B^*\} \). But by the definition of the source operator, we see that \( \sigma_{AB}(V_A \times V_B) = \sigma_A(V_A) \times \sigma_B(V_B) \). This way, the open set defined by \( \sigma_{AB} \) is \( \sigma_{AB}(V_A \times V_B)^C = (\sigma_A(V_A) \times \sigma_B(V_B))^C = \sigma_A(V_A)^C \times \sigma_B(V_B)^C \), but \( \sigma_A(V_A) \) and \( \sigma_B(V_B) \) are \( U_A \) and \( U_B \), respectively. Then \( \sigma_{AB}(V_A \times V_B)^C = U_A \times U_B \), which is an open set of the product topology. Since it is true for every subsets \( U_A \) and \( U_B \), the product topology coincides with the source topology of the resultant automaton.

When two automata are finite and independent, the number of states of the resultant automaton is simply the multiplication between the number of states of the two automata.
4.2 \( Q \)-dependent automata

We are going to call \( Q \)-dependent automata the ones which are related in a way that the set of states of one interfere in the set of states of another. Specifically, we mean that given two automata \( A_1 \) and \( A_2 \), there are states in \( A_1 \) that will be accessible only if the current state of \( A_2 \) is in some specific subset and this relation can be a two-hand relation.

In a generic way, looking into two automata with set of states \( Q_1 \) and \( Q_2 \), we can represent the situation in the following way:

\[
\begin{align*}
\text{\( Q_1 \)} & \\
U_1 & \cap V_1 \\
\text{\( Q_2 \)} & \quad \quad \quad \quad \quad \quad \quad \quad \\
U_2 & \cap V_2
\end{align*}
\]

\( U_1 \subset Q_1 \) is the restricted set of \( Q_1 \) and \( V_2 \subset Q_2 \) is the condition set of \( U_1 \). \( U_2 \subset Q_2 \) is the restricted set of \( Q_2 \) and \( V_1 \subset Q_1 \) is the condition set of \( U_2 \).

Notice the condition sets and restricted sets can have empty intersections or not. In addition, there can be more than one pair of restricted and conditions sets.

The logic is that for a given state \( q_0 \in Q_1 \) \( q_0 \notin U_1 \), a transition from \( q_0 \) for some \( q \in U_1 \) is only possible if the current state of \( Q_2 \) belongs to \( V_2 \). Also, the state \( q \) can continue in access only if the state of \( Q_2 \) still belonging to \( V_2 \).

The set of states of the resultant automaton would be \( Q_1 \times Q_2 \) if \( Q_1 \) and \( Q_2 \) were independents, but it is not the case now. We are going to work analyzing the set of states of the resultant automaton \( Q \), which elements are pairs \((q_1, q_2)\), such that \( q_1 \in Q_1 \) and \( q_2 \in Q_2 \).

The pairs in the form \((q_{u1}, q_2)\) \( q_{u1} \in U_1, q_2 \notin V_2 \) and \((q_1, q_{u2})\) \( q_{u2} \in U_2, q_1 \notin V_1 \) can’t occur, since the build of the relation between the sets. In this simpler case, we can write the set of states of the resultant automaton as \( Q_1 \times Q_2 - U_1 \times (Q_2 - V_2) - U_2 \times (Q_1 - V_1) \). In which \( U_1 \) is restricted by \( V_2 \) and \( U_2 \) is restricted by \( V_1 \).

**Definition 4.5.** Two automata \( A = (Q_A, \Gamma_A, \delta_A) \) and \( B = (Q_B, \Gamma_B, \delta_B) \) are called \( Q \)-dependent automata if, and only if, the resultant automaton is
expressed as $A \otimes B = (Q, \Gamma, \delta)$. Where 
$$Q = Q_A \times Q_B - U_1^A \times (Q_B - V_1^B) - U_2^A \times (Q_B - V_2^B) - ... - U_I^A \times (Q_A - V_I^A) - U_2^B \times (Q_A - V_2^A) - ...;$$

$U_1^A, U_2^A, ...$ are the restricted sets of $A$ and $U_1^B, U_2^B, ...$ are the restricted sets of $B$, while $V_1^A, V_2^A, ...$ are the condition sets of $A$ and $V_1^B, V_2^B, ...$ are the condition sets of $B$. Each condition set of $B$ is associated with one restricted set of $A$ and vice-versa. Specifically, $V_i^B$ is condition to $U_i^A$ and $V_j^A$ is condition to $U_j^B$.

$$\Gamma = \Gamma_A \times \Gamma_B;$$

If $q = (q_A, q_B) \mid q_A \in Q_A, q_B \in Q_B$ and $m = (m_A, m_B) \mid m_A \in \Gamma_A^*, m_B \in \Gamma_B^*$, then $\delta(q, m) = (\delta_A(q_A, m_A), \delta_B(q_B, m_B))$ if $\delta(q, m) \in Q$.

**Theorem 4.6.** Given two finite $\Gamma$-dependent automata $A$ and $B$, with set of states $Q_A$ and $Q_B$, respectively, restricted and condition sets named as in 4.5. The number of states of $A \otimes B$ is given by 
$$\eta(Q) = \eta(Q_A)\eta(Q_B) - \eta(U_1^A)\eta(Q_B) - \eta(V_1^B) - \eta(U_2^A)\eta(Q_B) - \eta(V_2^B) - ... - \eta(U_I^A)\eta(Q_B) - \eta(V_I^B) - \eta(U_1^B)\eta(Q_A) - \eta(V_1^A) - \eta(U_2^B)\eta(Q_A) - \eta(V_2^A) - ....$$ 

where $\eta(K)$ means number of elements of the set $K$.

**Proof.** The proof is a direct consequence of the multiplicative principle and the operations between the sets that define $Q$. One could verify it by using operations with sets in the definition of $Q$. \hfill \square

### 4.3 $\Gamma$-dependent automata

Intuitively, $\Gamma$-dependent automata can be seen as automata in which certain transitions in one of the automata can only be done if the transition happening in the other automaton is using some specific strings.

**Definition 4.7.** Let $A = (Q_A, \Gamma_A, \delta_A)$ and $B = (Q_B, \Gamma_B, \delta_B)$ two automata. We say that $A$ is $\Gamma$-**dependent** of $B$ if, and only if, the resultant automaton is given by $R = (Q, \Gamma, \delta)$, where 
$$Q = Q_A \times Q_B;$$

$$\Gamma = \Gamma_A \times \Gamma_B$$

and 
$$\delta(q, m) = (\delta_A(q_A, m_A), \delta_B(q_B, m_B))$$

with $q = (q_A, q_B)$ and $m = (m_A, m_B)$, along with the condition that for every pair $(q_i, m_j) \mid q_i \in Q_A, m_j \in \Gamma_A^*$, there is a set $\Gamma_{q_i, m_j} \subseteq \Gamma_B$ such that the transition $\delta(q, m) = (\delta_A(q_i, m_j), \delta_B(q_B, m_B))$ is only defined if $m_B \in \Gamma_{q_i, m_j}$.

Initially is very attractive the idea that a $\Gamma$-dependence association has no impact in the set of states of the resultant automaton, but this idea is not correct, as we can see with the following example.

We can think about two simple automata that can be analyzed by hand. Consider the automaton $A = (Q_A, \Gamma_A, \delta_A)$, with $Q_A = \{q_1, q_2\}$, $\Gamma_A = \{a, e\}$ in which $e$ is the empty string and the only defined transition using $a$ is $\delta_A(q_1, a) = q_2$. Let's look also to the automaton $B = (Q_B, \Gamma_B, \delta_B)$, such that $Q_B = \{q_1, q_2, q_3\}$, $\Gamma_B = \{b, \gamma, e\}$ and the following transitions are defined (despite the ones that involves $e$): $\delta_B(q_1, b) = q_2$ and $\delta_B(q_1, \gamma) = q_3$. 

19
Suppose \( A \) is \( \Gamma \)-dependent of \( B \) and the transition \( \delta_A(q_1, a) \) is defined only if the transition happening in \( B \) is by \( \gamma \). We can see the set of states \( Q \) of the resultant automaton won’t be simply \( Q_A \times Q_B \), since it is impossible to get the pair \( (q_2, q'_2) \). By a counter-example we showed that a \( \Gamma \)-dependence relation can change the final resultant automaton.

So, to describe the set of states of the resultant automaton, we are going to think in the general case of two automata \( A = (Q_A, \Gamma_A, \delta_A) \) and \( B = (Q_B, \Gamma_B, \delta_B) \).

The transition in \( \delta_A(q_i, m_j) \), being \( q_i \) the \( i \)-th element of \( Q_A \) and \( m_j \) the \( j \)-th element of \( \Gamma_A \), is associated with the set \( \Gamma_{B,i,j} \subseteq \Gamma_B \). Analogously, the transition \( \delta_B(q'_i, m'_j) \), where \( q'_i \) is the \( i \)-th element of \( Q_B \) and \( m'_j \) is the \( j \)-th element of \( \Gamma_B \), is associated with the set \( \Gamma_{A,i,j} \subseteq \Gamma_A \).

Calling \( Q_R \) the set of states of the resultant automaton, the conditions to some state \( q = (q_A, q_B) \) belong to \( Q_R \) are:

- For a \( q_i \mid \delta_A(q_i, m_j) = q_A, \exists \gamma_p \in \Gamma_{B,i,j} \mid \delta_B(q_k, \gamma_p) = q_B \) for some \( q_k \in Q_B \);
- \( m_j \in \Gamma_{B,k,p} \).

We also must apply the criteria to elements in \( Q_B \).

This way we describe the set \( Q_R \).

So a transition \( \delta(q, m) \), with \( q = (q_i, q_p) \) and \( m = (m_j, m_q) \), is defined if \( m_q \in \Gamma_{B,i,j} \) and \( m_p \in \Gamma_{A,p,q} \).

5 Conclusion

A study focused in the topological structure of the set of states of an automaton, based in its transitions generates an interesting amount of tools to analyze a state-machine or a system of state-machines.

We can realize that the development of the ideas related to the topology of an automaton can make easier the understanding or the description of state-machines in general. In addition, we presented concepts and theorems useful in the study and comparison of automata that have similarities, even if they look very distinct.

It was also possible to build a model of probabilistic automaton which makes viable the exploration of the transitions itself, by the computation of the involved probabilities.

We made possible to utilize ideas of set theory in order to build models of restrictions between automata, defining different relations and associations between different automata and building a resultant automaton which contains features of all the involved automata. This also makes possible to study some topological behaviour of the resultant automata and the associated automata.

The topological approach to analyze an automaton may not be enough to deal with all the problems involving state-machines, but it can be a powerful tool to simplify problems and systems. It is a good approach in order to transform
problems in equivalent situations where one could solve in some easier way, even allowing us to get a new problem and solve it using old methods.

References

[1] David E. BAVEL Zamir; MULLER. “Conectivity and Reversibility in Automata”. In: Journal of Association for Computing Machinery 18 (1970).

[2] Zamir BAVEL. “Source as a Tool in Automata”. In: Information and Control 17, 2 (1971), pp. 140–155.

[3] Zamir BAVEL. “Structure and Transition-Preserving Functions in Automata”. In: Journal of Association for Computing Machinery 15, 1 (1968).

[4] Pierre SCHAPIRA. General Topology. Paris, 2010.

[5] Arun K. SHUKLA Wagish; SRIVASTAVA. “A Topology for Automata II”. In: International Journal of Mathematics and Mathematical Science 9.3 (1986), pp. 425–428.

[6] Arun K. SHUKLA Wagish; SRIVASTAVA. “A Topology for Automata: A Note”. In: Information and Control 32 (1976), pp. 163–168.