Chapter VIII

An excursion into logic

1 Introduction.

This chapter is devoted to the presentation of the links between finite automata and logic. Büchi was the first to set up a logical language equivalent to finite automata. This logical formalism is rather uncommon since it makes use of second order variables. The reader will find in the notes at the end of this chapter the motivations that drove Büchi into this direction and a description of the genesis of this aspect of the theory of automata. We shall simply emphasize here two aspects of this question.

It is remarkable that all the logical theories occurring in this framework are decidable theories. This is a constant of the theory of finite automata in which most of the usual problems are not only decidable but even of low complexity. Showing off a decidable logical theory equivalent to finite automata is, in a sense, a confirmation of this general principle.

Furthermore, the logical formalism is the same for finite words and for infinite words. The only switch occurs in the interpretation of formulas, in contrast with the formalism of automata or rational expressions, in which the finite and the infinite case have to be distinguished. In a sense, the material of this chapter constitutes a justification of the unity of the theories of automata on finite or infinite words.

We first present in Section 2 the elements of formal logic necessary to understand the sequel. It consists essentially of very elementary notions which can be skipped by a reader having some background in logic. We introduce in particular monadic second order logic, which is the logical language equivalent with automata, as is shown further on. We detail the interpretation of formulas on words. It amounts mainly to consider a word as a structure by associating to each index i the letter in position i. The relations that are used are the order relation between the indexes and the relation $a(i)$ expressing that the letter in position i is an a.

In Section 3, we prove the equivalence between finite automata and monadic second order logic (Büchi's theorem). We prove this result for finite words and for infinite words. One of the consequences of this result is the decidability of the monadic second order logic on natural integers (also called monadic second order logic of one successor). Besides, the proof of this latter result is one the motivation of the work of Büchi. The extension of Büchi's decidability result to two successors is Rabin's theorem, which will be presented in Chapter X (Theorem X.4.3).

The conversion from automata to logic can be made by writing a formula stating that a word is the label of a successful path in an automaton. This proof brings us a supplementary clue: every formula of monadic second order logic is equivalent to a formula of low complexity in terms of quantifier alternation. In particular, the hierarchy based on the number of quantifier alternations collapses. We shall see later on that this is a major difference between first and second order.

We also investigate the expressive power of the logical language obtained by replacing the order relation on integers by the successor function. We shall see that, for infinite words, the monadic second order logics of these two languages are equivalent. It should not be a surprise to the reader familiar with logic, since the order relation is definable in monadic second order logic of the successor function.

Furthermore, we show that monadic second order logic is, on the integers, equivalent with *weak* monadic second order logic. The weak logic utilizes the same formulas as the "strong" one but the interpretation is more restrictive: set variables are always interpreted as finite sets of integers. The equivalence between weak and strong logic is a consequence of McNaughton's theorem. Indeed, one can deduce from this theorem that each rational set $X \subset A^\omega$ is a boolean combination of subsets of the form $\overrightarrow{Y}$ with $Y \subset A^*$. And such a set is definable by the formula

$$\forall x\ \exists y\ (y > x \wedge \varphi(y))$$

where $\varphi(y)$ is the formula expressing that the y's first letters of a word are in Y. Since Y is a set of finite words, we can restrict ourselves to weak logic to interpret $\varphi(y)$. This interdependence of results expressible in logic and in terms of automata is the mark of the deep and real connection between the two approaches. We shall see other examples in the sequel when we shall deal with first order logic.

In Section 4, we present the corresponding theory for first order logic. We start by first order theory of the linear order since, and it is a first difference with the monadic case, the logic of the linear order and that of the successor are no longer equivalent.

The first order logic of the linear order is shown to be equivalent to aperiodic automata or, as we have seen in Chapter VI, to star-free sets. This is true for finite words and for infinite words.

In Section 4.2, we describe the hierarchy of the first order logic of the linear order, corresponding to the alternate use of existential and universal quantifiers. We show that this hierarchy corresponds, on words, to the concatenation hierarchy described in Chapter VI. This leads to a doubly satisfying situation: first order formulas not only correspond globally to a natural class of recognizable sets (the star-free sets), but this correspondence holds level by level.

In Section 5, we study a more restricted language, the first order logic of the successor. We give an effective characterization of the sets of infinite words definable in this logic: they are the threshold locally testable sets (defined in Chapter VII) which are in the class Δ_2 (defined in Chapter III. We shall introduce on this occasion a technique of proof, called *Fraïssé-Ehrenfeucht* game, and we shall use it for proving the characterization theorem.

We introduce in the last section the formalism of *temporal logic* and show that this logic is equivalent to first order logic. Finally, we describe the expressive power of the temporal logic without the until operator.

In Chapters IX and X, we shall see how the logical framework presented in this chapter extends to the more general cases of bi-infinite words and infinite trees.

2 The formalism of logic.

In this section, we review the basic definitions of logic that will be used in this book. We shall define successively first order logic, second order, monadic second order and weak monadic second order logic.

2.1 Syntax.

We shall first define the syntax of logical formulas. Let us start by *first order logic*.

The basic ingredients are the *logical symbols* which encompass the logical connectives: $\wedge$ (and), $\vee$ (or), $\neg$ (not), $\rightarrow$ (implies), the equality symbol $=$, the *quantifiers* $\exists$ (there exists) and $\forall$ (for all), an infinite set of variables (most often denoted by x, y, z, or x_0, x_1, x_2,...) and parenthesis (to ensure legibility of the formulas).

In addition to these logical symbols, we make use of a set $\mathcal{L}$ of non logical symbols. These auxiliary symbols can be of three types: relation symbols (for instance $<$), function symbols (for instance min), or constant symbols (for instance 0, 1). Expressions are built from the symbols of $\mathcal{L}$ by obeying the usual rules of the syntax, then first order formulas are built by using the logical symbols, and are denoted by $F_1(\mathcal{L})$. We now give a detailed description of the syntaxic rules to obtain the logical formulas in three steps, passing successively, following the standard terminology, from terms to atomic formulas and subsequently to formulas.

We shall illustrate the formation rules by taking as an example

$$\mathcal{L} = \{ < , \min, 0\}$$

in which $<$ is a binary relation symbol, min is a two-variable function symbol and 0 is a constant.

We first define the set of $\mathcal{L}$-*terms*. It is the least set of expressions containing the variables, the constant symbols of $\mathcal{L}$ (if there are some) which is closed under the following formation rule: if $t_1, t_2, \ldots, t_n$ are terms and if f is a function symbol with n arguments, then the expression $f(t_1, t_2, \ldots, t_n)$ is a term. In particular, if $\mathcal{L}$ does not contain any function symbol, the only terms are the variables and the constant symbols. In our example, the following expressions are terms:

$$x_0 \qquad \min(0,0) \qquad \min(x_1, \min(0, x_2)) \qquad \min(\min(x_0, x_1), \min(x_1, x_2))$$

The *atomic formulas* are formulas either of the form

$$(t_1 = t_2)$$

where t_1 and t_2 are terms, or of the form

$$R(t_1, \ldots, t_n)$$

where $t_1, \ldots, t_n$ are terms and where R is a n-ary relation symbol of $\mathcal{L}$. On our example, the following expressions are atomic formulas:

$$(\min(x_1, \min(0, x_2)) = x_1) \qquad (\min(0, 0) < 0)$$
$$(\min(\min(x_0, x_1), \min(x_1, x_2)) < \min(x_3, x_1))$$

Notice that, in order to improve legibility, we didn't take literally the definition of atomic formulas: indeed, since $<$ is a symbol of binary relation, one should write $<(t_1, t_2)$ instead of $t_1 < t_2$.

Finally, the *first order formulas* on $\mathcal{L}$ form the least set of expressions containing the empty formula and the atomic formulas and closed under the following formation rules:

(i) If $(\varphi_i)_{i \in I}$ is a finite family of first order formulas, so are the expressions

$$(\bigwedge_{i \in I} \varphi_i) \quad \text{and} \quad (\bigvee_{i \in I} \varphi_i)$$

(ii) If φ and ψ are first order formulas, so are the expressions

$$\neg\varphi \quad \text{and} \quad (\varphi \to \psi)$$

(iii) If φ is a first order formula and if x is a variable, then the expressions

$$(\exists x \varphi) \quad \text{and} \quad (\forall x \varphi)$$

are first order formulas.

To make notations easier, we set

$$\mathbf{true} = \bigwedge_{i \in \emptyset} \varphi_i \quad \text{and} \quad \mathbf{false} = \bigvee_{i \in \emptyset} \varphi_i$$

The following expressions are first order formulas of our example language:

$$(\exists x\ (\forall y\ ((y < \min(z, 0)) \wedge (x < 0)))) \qquad\qquad (\forall x\ (y = x))$$

Again, it is convenient to simplify notation by suppressing some of the parenthesis and we shall write the previous formulas under the form

$$\exists x\ \forall y\ (y < \min(x, 0)) \wedge (z < 0) \qquad\qquad \forall x\ y = x$$

In a first order formula, some variables occur after a quantifier (existential or universal): the occurrences of these variables are said to be *bounded* and the other occurrences are said to be *free*. For example, in the formula

$$\exists x\ (\underline{\underline{y}} < h(\underline{x}, 0)) \wedge \forall y\ (\underline{\underline{z}} < \underline{y})$$

the simply underlined occurrences of x and y are bounded and the occurrences of z and y doubly underlined are free. A variable is free if at least one of its occurrences is free. The set $FV(\varphi)$ of free variables of a formula φ can be defined inductively as follows:

(1) If φ is an atomic formula, $FV(\varphi)$ is the set of variables occurring in φ,
(2) $FV(\neg\varphi) = FV(\varphi)$
(3) $FV(\bigwedge_{i\in I} \varphi_i) = FV(\bigvee_{i\in I} \varphi_i) = \bigcup_{i\in I} FV(\varphi_i)$
(4) $FV(\varphi \to \psi) = FV(\varphi) \cup FV(\psi)$
(5) $FV(\exists x\varphi) = FV(\forall x\varphi) = FV(\varphi) \setminus \{x\}$

A *statement* is a formula in which all occurrences of variables are bounded. For example, the formula

$$\exists x\ \forall y\ (y < \min(x, 0))$$

is a statement.

We shall denote by $\varphi(x_1, x_2, \ldots, x_n)$ a formula φ in which the set of free variables is contained in $\{x_1, \ldots, x_n\}$ (but is not necessarily equal to $\{x_1, \ldots, x_n\}$).

The variables used in first order logic, or *first order*) are interpreted, as we shall see, as the elements of a set. In *second order logic*, one makes use of another type of variables, called *second order variables*, which represent relations. These variables are denoted traditionally by capital letters: X_0, X_1, etc.. One built in this way the set of second order formulas on $\mathcal{L}$, denoted by $F_2(\mathcal{L})$. The set of terms is the same as for first order. The *atomic formulas* are either of the form

$$(t_1 = t_2)$$

where t_1 and t_2 are terms, or of the form

$$R(t_1, \ldots, t_n) \qquad \text{or} \qquad X(t_1, \ldots, t_n)$$

where $t_1, \ldots, t_n$ are terms, R is a n-ary relation symbol of $\mathcal{L}$ and X is a variable representing a n-ary relation.

Finally, *second order formulas* on $\mathcal{L}$ form the least set of expressions containing the atomic formulas and closed under the following formation rules:

(i) If φ and ψ are second order formulas, then so are

$$\neg\varphi, \quad (\varphi \wedge \psi), \quad (\varphi \vee \psi), \quad (\varphi \to \psi)$$

(ii) If φ is a second order formula, if x is a variable and if X is a variable of relation, then the expressions

$$(\exists x\varphi) \qquad (\forall x\varphi) \qquad (\exists X\varphi) \qquad (\forall X\varphi)$$

are second order formulas.

We call *monadic second order logic* the fragment of second order logic in which the only relation variables are unary relation variables, in other words, variables representing subsets of the domain. By convenience, they are called *set variables*. We denote by $MF_2(\mathcal{L})$ the set of monadic second order logic on $\mathcal{L}$. We shall also use the notation $x \in X$ instead of $X(x)$.

2.2 Semantics.

We have adopted so far a syntactic point of view to define formulas with no reference to semantics. But of course, formulas would be uninteresting if they were meaningless. Giving a precise meaning to formulas requires to specify the domain on which we want to interpret each of the symbols of the language $\mathcal{L}$. Formally, a *structure* $\mathcal{S}$ on $\mathcal{L}$ is given by a nonempty set D, called *domain* and by a map defined on $\mathcal{L}$, called an *assignment* which associates

(1) to each n-ary relation symbol of $\mathcal{L}$, a n-ary relation defined on D,
(2) to each n-ary function symbol f of $\mathcal{L}$, a n-ary function defined on D,
(3) to each constant symbol c of $\mathcal{L}$, an element of D.

To decongest notations, we shall use the same notation for the relation (resp. function, constant) symbols and for the relations (resp. functions, constants) represented by these symbols. The context will allow us to determine easily what the notation stands for. For example, we shall always employ the symbol $<$ to designate the usual order relation on a set of integers, independently of the domain ($\mathbb{N}$, $\mathbb{Z}$, or a subset of $\mathbb{N}$).

We still have to interpret variables. Let us start by first order variables. Given a fixed structure $\mathcal{S}$ on $\mathcal{L}$, with domain D, a *valuation* on $\mathcal{S}$ is a map ν from the set of variables into the set D. It is then easy to extend ν into a function of the set of terms of $\mathcal{L}$ into D, by induction on the formation rules of terms:

(1) If c is a constant symbol, we put $\nu(c) = c$,
(2) if f is a n-ary function symbol and if $t_1, \ldots, t_n$ are terms,

$$\nu\big(f(t_1, \ldots, t_n)\big) = f(\nu(t_1) \ldots \nu(t_n))$$

If ν is a valuation and a an element of D, we denote by $\nu\binom{a}{x}$ the valuation ν' defined by

$$\nu'(y) = \begin{cases} \nu(y) & \text{if } y \neq x \\ a & \text{if } y = x \end{cases}$$

The notion of interpretation can be now easily formalized. Define, for each first order formula φ and for each valuation ν, the expressions "the valuation ν satisfies φ in $\mathcal{S}$", or "$\mathcal{S}$ satisfies $\varphi[\nu]$", denoted by $\mathcal{S} \models \varphi[\nu]$, as follows:

(1) $\mathcal{S} \models (t_1 = t_2)[\nu]$	if and only if $\nu(t_1) = \nu(t_2)$
(2) $\mathcal{S} \models R(t_1, \ldots, t_n)[\nu]$	if and only if $\big(\nu(t_1), \ldots, \nu(t_n)\big) \in R$
(3) $\mathcal{S} \models \neg\varphi[\nu]$	if and only if not $\mathcal{S} \models \varphi[\nu]$
(4) $\mathcal{S} \models (\bigwedge_{i \in I} \varphi)[\nu]$	if and only if for each $i \in I$, $\mathcal{S} \models \varphi_i[\nu]$
(5) $\mathcal{S} \models (\bigvee_{i \in I} \varphi)[\nu]$	if and only if there exists $i \in I$, $\mathcal{S} \models \varphi_i[\nu]$
(6) $\mathcal{S} \models (\varphi \rightarrow \psi)[\nu]$	if and only if $\mathcal{S} \not\models \varphi[\nu]$ or $\mathcal{S} \models \psi[\nu]$
(7) $\mathcal{S} \models (\exists x\varphi)[\nu]$	if and only if $\mathcal{S} \models \varphi[\nu\binom{a}{x}]$ for some $a \in D$
(8) $\mathcal{S} \models (\forall x\varphi)[\nu]$	if and only if $\mathcal{S} \models \varphi[\nu\binom{a}{x}]$ for each $a \in D$

Note that, actually, the truth of the expression "the valuation ν satisfies φ in $\mathcal{S}$" only depends on the values taken by the free variables of φ. In particular, if φ is a statement, the choice of the valuation is irrelevant. Therefore, if φ is a statement, one says that φ is satisfied by $\mathcal{S}$ (or that $\mathcal{S}$ satisfies φ), and denote by $\mathcal{S} \models \varphi$, if, for each valuation ν, $\mathcal{S} \models \varphi[\nu]$.

Next we move to the interpretation of second order formulas. Given a structure $\mathcal{S}$ on $\mathcal{L}$, with domain D, a *second order valuation* on $\mathcal{S}$ is a map ν which associates to each first order variable an element of D and to each n-ary relation variable a subset of D^n (i.e. a n-ary relation on D).

If ν is a valuation and R a subset of D^n, $\nu\binom{R}{X}$ denotes the valuation ν' defined by

$$\nu'(x) = \nu(x) \text{ if } x \text{ is a first order variable}$$
$$\nu'(Y) = \begin{cases} \nu(Y) & \text{if } Y \neq X \\ R & \text{if } Y = X \end{cases}$$

The notion of interpretation, already defined for first order, is supplemented by the following rules:

(9) $\mathcal{S} \models (X(t_1, \ldots, t_n))[\nu]$ if and only if $\big(\nu(t_1), \ldots, \nu(t_n)\big) \in \nu(X)$

(10) $\mathcal{S} \models (\exists X \varphi)[\nu]$ if and only if there exists $R \subseteq D^n$, $\mathcal{S} \models \varphi[\nu\binom{R}{X}]$

(11) $\mathcal{S} \models (\forall X \varphi)[\nu]$ if and only if for each $R \subseteq D^n$, $\mathcal{S} \models \varphi[\nu\binom{R}{X}]$

Weak monadic second order logic is composed with the same formulas than monadic second order logic, but the interpretation is even more restricted: are only considered valuations which associate to set variables *finite* subsets of the domain D.

Two formulas φ and ψ are said to be *logically equivalent* if, for each structure $\mathcal{S}$ on $\mathcal{L}$, we have $\mathcal{S} \models \varphi$ if and only if $\mathcal{S} \models \psi$.

It is easy to see that the following formulas are logically equivalent:

(1)	$\varphi \wedge \psi$	and	$\neg(\neg\varphi \vee \neg\psi)$
(2)	$\varphi \rightarrow \psi$	and	$\neg\varphi \vee \psi$
(3)	$\forall x \varphi$	and	$\neg(\exists x \, \neg\varphi)$
(4)	$\varphi \vee \psi$	and	$\psi \vee \varphi$
(5)	$\varphi \wedge \psi$	and	$\psi \wedge \varphi$
(6)	$\varphi \wedge false$	and	$false$
(7)	$\varphi \vee false$	and	φ

Consequently, up to logical equivalence, we may assume that the formulas are built without the symbols $\wedge$, $\rightarrow$ and $\forall$.

Logical equivalence also permits to give a more structured form to formulas. A formula is said to be under *disjunctive normal form* if it can be written as disjunction of conjunctions of atomic formulas or of negations of atomic formulas, in other words under the form

$$\bigvee_{i \in I} \bigwedge_{j \in J_i} (\varphi_{ij} \vee \neg\psi_{ij})$$

where I and the J_i are finite sets, and where the φ_{ij} and the ψ_{ij} are atomic formulas. The next result is standard and easily proved.

Proposition 2.1 *Every quantifier free formula is logically equivalent with a quantifier free formula in disjunctive normal form.*

One can set up a hierarchy inside first order formulas as follows. Let $\Sigma_0 = \Pi_0$ the set quantifier-free formulas. Next, for each $n \geq 0$, denote by Σ_{n+1} the least set Δ of formulas such that

(1) Δ contains the boolean combinations of formulas of Σ_n,
(2) Δ is closed under disjunctions and finite conjunctions,
(3) if $\varphi \in \Delta$ and if x is a variable, $\exists x\varphi \in \Delta$.

Similarly, denote by Π_{n+1} the least set Γ of formulas such that
(1) Γ contains the boolean combinations of formulas of Π_n,
(2) Γ is closed under disjunctions and finite conjunctions,
(3) if $\varphi \in \Gamma$ and if x is a variable, $\forall x\varphi \in \Gamma$.

In particular, Σ_1 is the set of *existential* formulas — that is of the form

$$\exists x_1 \, \exists x_2 \, \ldots \, \exists x_k \, \varphi$$

where φ is quantifier-free.
Finally, we denote by $\mathcal{B}\Sigma_n$ the set of boolean combinations of formulas of Σ_n.
A first order formula is said to be in *prenex normal form* if it is of the form

$$\psi = Q_1 x_1 \; Q_2 x_2 \; \ldots \; Q_n x_n \; \varphi$$

where the Q_i are existential or universal quantifiers ($\exists$ or $\forall$) and φ is quantifier-free. The sequence $Q_1 x_1 \; Q_2 x_2 \; \ldots \; Q_n x_n$, which can be considered as a word on the alphabet

$$\{\ \exists x_1, \exists x_2, \ldots, \forall x_1, \forall x_2, \ldots\ \},$$

is called the quantification prefix of ψ. The interest of these formulas in prenex normal form comes from the following result.

Proposition 2.2 *Every first order formula is logically equivalent to a formula in prenex normal form.*

Proof. It suffices to verify that if the variable x does not occur in the formula ψ then

$$\exists x(\varphi \wedge \psi) \equiv (\exists x\varphi) \wedge \psi$$
$$\exists x(\varphi \vee \psi) \equiv (\exists x\varphi) \vee \psi$$

and the same formulas hold for the quantifier $\forall$. Hence it is possible, by renaming the variables, to throw back the quantifiers to the outside. $\square$

Proposition 2.2 can be improved to take into account the level of the formula in the Σ_n hierarchy.

Proposition 2.3 *For each integer $n \geq 0$,*
(1) *Every formula of Σ_n is logically equivalent to a formula in prenex normal form in which the quantification prefix is a sequence of n (possibly empty) alternating blocks of existential and universal quantifiers, starting with a block of existential quantifiers.*
(2) *Every formula of Π_n is logically equivalent to a formula in prenex normal form in which the quantification prefix is a sequence of n (possibly empty) alternating blocks of existential and universal quantifiers, starting with a block of universal quantifiers.*

For example, the formula

$$\underbrace{\exists x_1\ \exists x_2\ \exists x_3}_{\text{block 1}}\ \underbrace{\forall x_4\ \forall x_5}_{\text{block 2}}\ \underbrace{\exists x_6\ \exists x_7}_{\text{block 3}}\ \varphi(x_1, \ldots, x_6)$$

belongs to Σ_3 (and also to all Σ_n's such that $n \geq 3$). Similarly the formula

$$\underbrace{\quad}_{\text{block 1}}\ \underbrace{\forall x_4\ \forall x_5}_{\text{block 2}}\ \underbrace{\exists x_6\ \exists x_7}_{\text{block 3}}\ \varphi(x_1, \ldots, x_7)$$

belongs to Σ_3 and to Π_2, but not to Σ_2, since the counting of blocks of a Σ_n-formula should always begin by a possibly empty block of existential quantifiers.

One can introduce normal forms and a hierarchy for second order monadic formulas. Thus, one can show that every monadic second order formula is logically equivalent to a formula of the form

$$\psi = Q_1 X_1\ Q_2 X_2\ \ldots\ Q_n X_n\ \varphi$$

where the Q_i are existential or universal quantifiers and φ is a first order formula.

2.3 Logic on words.

The logical language that we shall use now was introduced by Büchi under the name of "sequential calculus". It permits to formalize certain properties of words. To interpret formulas on words, we shall consider a word $u = u(0)u(1)\ldots$ as a map associating a letter to each index. The set of indexes will itself be considered as a subset of the set

$$\mathcal{N} = \mathbb{N} \cup \{\infty\}$$

the set of natural numbers with a new maximal element, denoted by ∞.

Let A be an alphabet. For an infinite word $u \in A^\omega$, we set $|u| = \infty$, so that the length of a word is always an element of the set $\mathcal{N}$. For each word $u \in A^\infty$, one defines the *domain* of u, denoted by $\mathrm{Dom}(u)$ as

$$\mathrm{Dom}(u) = \{i \in \mathcal{N} \mid 0 \leq i \leq |u|\}$$

So, if u is a finite word, $\mathrm{Dom}(u) = \{0, \ldots, |u|\}$ and if u is an infinite word, $\mathrm{Dom}(u) = \mathcal{N}$. Define for each letter $a \in A$ a unary relation $\mathbf{a}$ on the domain of u by

$$\mathbf{a} = \{i < |u| \mid u(i) = a\}.$$

Finally, let us associate to each word u the structure

$$\mathcal{M}_u = \big(\mathrm{Dom}(u), (\mathbf{a})_{a \in A}\big),$$

Beware that, with these definitions, the domain of a finite or infinite word comprises a supplementary position at the end. One can imagine that this position is occupied by an end symbol (the \$ prized by lexical analyzers...).

For example, if $u = abbaab$, then $\mathrm{Dom}(u) = \{0, 1, \ldots, 6\}$, $\mathbf{a} = \{0, 3, 4\}$ and $\mathbf{b} = \{1, 2, 5\}$. If $u = (aba)^\omega$, then

$$\mathbf{a} = \{n \in \mathbb{N} \mid n \equiv 0 \mod 3 \text{ or } n \equiv 2 \mod 3\} \text{ and}$$
$$\mathbf{b} = \{n \in \mathbb{N} \mid n \equiv 1 \mod 3\}$$

From now and on, we shall interpret logical formulas on words, that is, on a structure of the form $\mathcal{M}_u$ as explained above. Let φ be a statement. A finite or infinite word $u \in A^\infty$ *satisfies* φ if the structure $\mathcal{M}_u$ satisfies φ. This is denoted by $u \models \varphi$. We also say that u is a model of φ. Define the *spectrum* of φ as the set

$$S(\varphi) = \{u \in A^\infty \mid u \text{ satisfies } \varphi\}$$

We also set $S^*(\varphi) = S(\varphi) \cap A^*$, $S^+(\varphi) = S(\varphi) \cap A^+$ and $S^\omega(\varphi) = S(\varphi) \cap A^\omega$.

From now and on, all the variables will be interpreted as natural integers or ∞. Therefore, we shall use logical equivalence restricted to interpretations of domain $\mathcal{N}$.

The various logical languages that we shall consider all contain, for each $a \in A$, a unary relation symbol denoted $(\mathbf{a})$ when no confusion arises. We shall also use two other non logical symbols, $<$ and S, that will be interpreted as the usual order and as the successor relation on $\mathrm{Dom}(u)$: $S(x, y)$ if and only if $y = x + 1$. In the sequel, we shall mainly consider two logical languages: the language

$$\mathcal{L}_< = \{<\} \cup \{\mathbf{a} \mid a \in A\}$$

will be called the *language of the linear order* and the language

$$\mathcal{L}_S = \{S\} \cup \{\mathbf{a} \mid a \in A\}$$

will be called the *language of the successor*. The atomic formulas of the language of the linear order are of the form

$$\mathbf{a}(x), \qquad x = y, \qquad x < y$$

and those of the language of the successor are of the form

$$\mathbf{a}(x), \qquad x = y, \qquad S(x, y).$$

We shall denote respectively by $F_1(<)$ and $MF_2(<)$ the set of first order and monadic second order formulas of signature $\{<, (\mathbf{a})_{a\in A}\}$. Similarly, we denote by $F_1(S)$ and $MF_2(S)$ the same sets of formulas of signature $\{S, (\mathbf{a})_{a\in A}\}$. Inside $F_1(<)$ (resp. $F_1(S)$), the $\Sigma_n(<)$ (resp. $\Sigma_n(S)$) hierarchy is based on the number of quantifier alternations.

We shall now start the comparison between the various logical languages we have introduced. First of all, the distinction between the signatures S and $<$ is only relevant for first order in view of the following proposition. A relation $R(x_1, \cdots, x_n)$ on the integers is said to be *defined* by a formula $\varphi(x_1, \ldots, x_n)$ if, for each $i_1, \ldots, i_n \geq 0$, $R(i_1, \ldots, i_n)$ if and only if $\varphi(i_1, \ldots, i_n)$ is true.

Proposition 2.4 *The successor relation can be defined in $F_1(<)$, and the order relation on integers can be defined in $MF_2(S)$.*

Proof. The successor relation can be defined by the formula

$$(i < j) \wedge \forall k \Big((i < k) \rightarrow ((j = k) \vee (j < k))\Big)$$

which states that $j = i + 1$ if i is smaller than j and there exist no element between i and j. Conversely, $i < j$ can be expressed in $MF_2(S)$ as follows:

$$\exists X \Big[\forall x \forall y \Big(((x \in X) \wedge S(x, y)) \rightarrow (y \in X)\Big)\Big] \wedge (j \in X) \wedge (i \notin X)$$

which intuitively means that there exists an interval of the form $[k, +\infty[$ containing j but not i. $\square$

On the contrary, we shall see later on (Corollary 5.10) that the relation $<$ cannot be defined in $F_1(S)$.

Given two sets of formulas F and G, we use the notation $F \leq G$ if for each formula of F, there exists an equivalent formula in G (we remind the reader that the variables are always interpreted as integers). Denote by $F \equiv G$ the associated equivalence and by $F < G$ the relation $F \leq G$ but $F \not\equiv G$. The next result summarizes the expressive power of the various logical languages.

Proposition 2.5 *The following relations hold:*

$$F_1(S) < F_1(<) < MF_2(S) = MF_2(<)$$

Proof. Proposition 2.4 shows the double inequality $F_1(S) \leq F_1(<) \leq MF_2(S)$. The fact that these inequalities are strict follow respectively from Corollary 5.10 and Corollary 4.4, which will be proved later on.

Finally, we have $MF_2(S) \leq MF_2(<)$ since S can be expressed in $F_1(<)$, which in turn is contained in $MF_2(<)$. But we also have $MF_2(<) \leq MF_2(S)$ since, by Proposition 2.4, the relation $<$ can be defined in $MF_2(S)$. $\square$

It will be convenient to enrich our logical languages by adding auxiliary formulas which can be considered as abbreviations. We shall see an example a few lines below with the predicate $\mathrm{Finite}(X)$ which expresses the fact that a set (of integers) is finite.

If x is a variable, it is convenient to write $x + 1$ (resp. $x - 1$) to replace a variable y submitted to the condition $S(x, y)$ (resp. $S(y, x)$). However, the reader should beware of the fact that $x + y$ is not definable in $MF_2(<)$ (Exercise 3).

We can also use the symbols $\leq$ and $\neq$ with their usual interpretations: $x \leq y$ stands for $(x < y) \vee (x = y)$ and $x \neq y$ for $\neg(x = y)$.

Another example, to be used below, are the symbols min, max, which designate the first and the last element of the domain. These symbols can be defined in $F_1(S)$ with two alternations of quantifiers:

$$\begin{aligned} \min : &\quad \exists \min\ \forall x\ \neg S(x, \min) \\ \max : &\quad \exists \max\ \forall x\ \neg S(\max, x) \end{aligned}$$

These symbols permit to express that u is the empty word ($\max = \min$) or that u is finite ($(\max = \min) \vee \exists\, x S(x, \max)$). We shall denote by FINITE the latter formula.

We shall sometimes need a parametrized, or relative, notion of satisfaction for a formula. Let indeed $\varphi(x, y)$ be a formula with only two free variables x and y. Let $u \in A^\infty$ be a word and $i, j \in \mathrm{Dom}(u)$. A word $u \in A^\infty$ is said to satisfy the formula φ between i and j if $u_i \dots u_{j-1} \models \varphi$ (or $u_i u_{i+1} \dots \models \varphi$ if $j = \infty$).

Proposition 2.6 *For each statement φ, there exists a formula $\varphi(x, y)$ with the same signature, the same order (and, in the case of a first order formula, the* *same level in the hierarchy Σ_n), having x and y as unique free variables and which satisfies the following property: for each finite or infinite word u and for each $s, t \in \mathrm{Dom}(u)$, $u \models \varphi(s,t)$ if and only if u satisfies φ between s and t.*

Proof. The formulas $\varphi(x,y)$ are built by induction on the formation rules as follows: if φ is an atomic formula, we set $\varphi(x,y) = \varphi$. Otherwise, we set

$$\begin{aligned}
(\neg\varphi)(x,y) &= \neg\varphi(x,y)\\
(\varphi \vee \psi)(x,y) &= \varphi(x,y) \vee \psi(x,y)\\
(\exists z\varphi)(x,y) &= \exists z \ ((x \leq z) \wedge (z < y) \wedge \varphi(x,y))\\
(\exists X\varphi)(x,y) &= \exists X \ \ ((\forall\ z(X(z) \to (x \leq z) \wedge (z < y)) \wedge \varphi(x,y))
\end{aligned}$$

It is easy to verify that the formulas $\varphi(x,y)$ built in this way have the required properties. □

We conclude this section by examining the place of the weak theories of the linear order and the successor. First of all, one can verify that the formulas used in the proof of Proposition 2.4 actually define the relation $<$ in $WMF_2(S)$ (Exercise 1). It follows that

$$WMF_2(S) = WMF_2(<)$$

and that the weak monadic theories of the successor and of the linear order are the same.

We shall see on the other hand that

$$WMF_2(<) = MF_2(<)$$

so that the weak and strong theories of the linear order coincide. We can also note immediately the next result, which is true in a much more general setting (actually as soon as one can write a formula expressing that a set is finite).

Proposition 2.7 *The formula $WMF_2(<) \leq MF_2(<)$ holds.*

Proof. One can write in $MF_2(<)$ a formula $\mathrm{Finite}(X)$ expressing that a set X is finite. For instance

$$\mathrm{Finite}(X) = \exists x\ (\forall y\ X(y) \to y < x)$$

Every formula of $WMF_2(<)$ is equivalent to a formula of the form

$$\psi = Q_1X_1 \ldots Q_nX_n\varphi$$

where φ is a first order formula. The formula ψ is then equivalent to the formula

$$Q_1X_1 \ldots Q_nX_n\varphi \wedge \bigwedge_{1\leq i\leq n} \mathrm{Finite}(X_i)$$

of $MF_2(<)$. □

The formulas in which the predicates **a** do not occur pertain uniquely to integers. We shall see in the next section that the words have nevertheless the same role to play, as soon as formulas with free variables have to be interpreted.

3 Monadic second order logic on words.

This section is devoted to the proof of a result of Büchi stating that the subsets of A^∞ definable in monadic second order logic are exactly the rational (resp. ω-rational) sets.

Theorem 3.1 *Let $X \subset A^\infty$ be a set of finite or infinite words. The following conditions are equivalent:*
(1) *X is definable by a formula of $MF_2(<)$,*
(2) *X is recognizable.*

The proof of this result can be decomposed into two parts: passing from words to formulas, and from formulas to words.

To pass from words to formulas, we simulate the behaviour of an automaton by a formula.

Proposition 3.2 *For each Büchi automaton $\mathcal{A} = (Q, A, E, I, F)$, there exist formulas $\varphi_+, \varphi_\omega$ of $MF_2(<)$ such that $S^+(\varphi_+) = L^+(\mathcal{A})$, $S^\omega(\varphi_\omega) = L^\omega(\mathcal{A})$ and $L^\infty(\mathcal{A}) = S(\varphi_+ \vee \varphi_\omega)$.*

Proof. Suppose that $Q = \{1, \ldots, n\}$. We first write a formula ψ expressing the existence of a path of label u. To this purpose, we associate to each state q a set variable X_q which encodes the set of positions in which a given path visits the state q. The formula states that the X_q's are pairwise disjoint and that if a path is in state q in position x, in state q' in position $x+1$ and if the x-th letter is an a, then $(q, a, q') \in E$. This gives

$$\psi = \Big(\bigwedge_{q \neq q'} \neg\exists x (X_q(x) \wedge X_{q'}(x))\Big) \wedge \\ \Big(\forall x \forall y \; S(x,y) \rightarrow \bigvee_{(q,a,q') \in E} (X_q(x) \wedge \mathbf{a}(x) \wedge X_{q'}(y))\Big)$$

It remains to state that the path is successful. For a finite path, it suffices to know that 0 belongs to one of the X_q's such that $q \in I$ and that max belongs to one of the X_q's such that $q \in F$. Therefore, we set

$$\psi_+ = \psi \wedge \mathrm{FINITE} \wedge \Big(\bigvee_{q \in I} X_q(0)\Big) \wedge \Big(\bigvee_{q \in F} X_q(\max)\Big)$$

An infinite path is successful if 0 belongs to one of the X_q's such that $q \in I$ and if, for each position x, there exists a position $y > x$ in which the path visits a final state. In summary, we set:

$$\psi_\omega = \psi \wedge \Big(\bigvee_{q \in I} X_q(0)\Big) \wedge \Big(\bigvee_{q \in F} \forall x \; \exists y \; (x < y \;\wedge\; X_q(y))\Big)$$

The formulas

$$\varphi_+ = \exists X_1 \exists X_2 \ldots \exists X_n (\psi_+)$$
$$\varphi_\omega = \exists X_1 \exists X_2 \ldots \exists X_n (\psi_\omega)$$

now entirely encode the automaton. □

To pass from statements to sets of words, a natural idea is to argue by induction on the formation rules of formulas. The problem is that the set $S(\varphi)$ is only defined when φ is a statement. The traditional solution in this case consists of adding constants to interpret free variables to the structure in which the formulas are interpreted. For the sake of homogeneity, we proceed in a slightly different way, so that these structures remain words.

The idea is to use an extended alphabet of the form

$$B_{p,q} = A \times \{0,1\}^p \times \{0,1\}^q$$

such that p (resp. q) is greater than or equal to the number of first order (resp. second order) variables of φ. An infinite word on the alphabet $B_{p,q}$ can be identified with the sequence

$$(u_0, u_1, \ldots, u_p, u_{p+1}, \ldots, u_{p+q})$$

where $u_0 \in A^\omega$ and $u_1, \ldots, u_p, u_{p+1}, \ldots, u_{p+q} \in \{0,1\}^\omega$. We are actually interested in the set $K_{p,q}$ of words of $B^\infty_{p,q}$ in which each of the components $u_1, \ldots, u_p$ contain exactly one occurrence of 1. If the context permits, we shall note B instead of $B_{p,q}$ and K instead of $K_{p,q}$. We also set

$$K_* = K \cap B^*, \qquad K_\omega = K \cap B^\omega$$

For example, if $A = \{a, b\}$, a word of $B^\omega_{3,2}$ is represented in Figure 3.1.

u_0	a	b	a	a	b	a	b	$\cdots$
u_1	0	1	0	0	0	0	0	$\cdots$
u_2	0	0	0	0	1	0	0	$\cdots$
u_3	1	0	0	0	0	0	0	$\cdots$
u_4	0	1	1	0	0	1	1	$\cdots$
u_5	1	1	0	1	0	1	0	$\cdots$

Figure 3.1. A word of $B^\omega_{3,2}$.

The elements of K are called *marked words* on A. This terminology expresses the fact that the elements of K are (finite or infinite) words in which labels marking certain positions have been added. Each of the p first rows only marks one position and the last q ones an arbitrary number of positions.

Recall that the *star-free* subsets of B^∞ are obtained from the sets B^∞ and $\emptyset$ by using the boolean operations and the *marked product*

$$(X, b, Y) \to XbY$$

which associates to $X \subset A^*$, $b \in B$ and $Y \subset B^\infty$, the subset XbY of B^∞. Star-free sets are in particular rational subsets of B^∞. Then we have the following property.

Proposition 3.3 *For each $p, q \geq 0$, the set $K_{p,q}$ is a star-free, and hence rational, subset of $B^\infty_{p,q}$.*

Proof. Set, for $1 \leq i \leq p$,

$$C_i = \{ (b_0, b_1, \ldots, b_{p+q}) \in B \mid b_i = 1 \}$$

Then K is the set of words of B^∞ containing exactly one letter of each C_i, for $1 \leq i \leq p$. Now the formula

$$K = \bigcap_{1 \leq i \leq p} B^* C_i B^\infty \setminus \bigcup_{1 \leq i \leq p} B^* C_i B^* C_i B^\infty.$$

shows that K is a star-free subset of B^∞. □

The interpretation of formulas on the words of $B^\infty_{p,q}$ follows the main lines of the interpretation described in Section 2.3, but the interpretation of $\mathbf{a}$ is slightly modified by setting

$$\mathbf{a} = \{ i < |u| \mid u_0(i) = a \}.$$

Let $\varphi(x_1, \ldots, x_r, X_1, \ldots, X_s)$ be a formula in which the first (resp. second) order free variables are $x_1, \ldots, x_r$ (resp. $X_1, \ldots, X_s$), with $r \leq p$ and $s \leq q$. Let $u = (u_0, u_1, \ldots, u_{p+q})$ be a word of $K_{p,q}$ and, for $1 \leq i \leq p$, denote by n_i the position of the unique 1 of the word u_i. On the example above, one would have $n_1 = 1$, $n_2 = 4$ and $n_3 = 0$. A word u is said to satisfy φ if u_0 satisfies $\varphi[\nu]$, where ν is the valuation defined by

$$\begin{aligned} &\nu(x_j) = n_j \qquad (1 \leq j \leq r) \\ &\nu(X_j) = \{ i \in \mathbb{N} \mid u_{p+j,i} = 1 \} \qquad (1 \leq j \leq s) \end{aligned}$$

In other words, each X_j is interpreted as the set of positions of 1's in u_{p+j}, and each x_j as the unique position of 1 in u_j. Note that for $p = q = 0$, we recover the customary interpretation of statements.

Set

$$S_{p,q}(\varphi) = \{ u \in K_{p,q} \mid u \text{ satisfies } \varphi(x_1, \ldots, x_p, X_1, \ldots, X_q) \}$$

Again, we shall sometimes simply use the notation $S(\varphi)$ instead of $S_{p,q}(\varphi)$. We also note that $S^*(\varphi) = S(\varphi) \cap A^*$ and $S^\omega(\varphi) = S(\varphi) \cap A^\omega$. Conjunctions and disjunctions are easily converted into boolean operations.

Proposition 3.4 *For each finite family of formulas $(\varphi_i)_{i \in I}$, the following equalities hold:*

(1) $S(\bigvee_{i \in I} \varphi_i) = \bigcup_{i \in I} S(\varphi_i)$,
(2) $S(\bigwedge_{i \in I} \varphi_i) = \bigcap_{i \in I} S(\varphi_i)$,
(3) $S(\neg\varphi) = K_{p,q} \setminus S(\varphi)$

Proof. This is an immediate consequence of the definitions. □

To conclude the proof of Theorem 3.1, it remains to prove by induction on the formation rules of formulas that the sets $S(\varphi)$ are rational. Let us start with the atomic formulas. As for the set K, we prove a slightly more precise result that will be used later on in this chapter.

Proposition 3.5 *For each variable x, y, for each set variable X and for each letter $a \in A$, the sets of the form $S(\mathbf{a}(x))$, $S(x = y)$, $S(x < y)$ and $S(X(x))$ are star-free, and hence rational, subsets of B^∞.*

Proof. Set, for $i, j \in \{1, \ldots, p+q\}$

$$\begin{aligned}
C_{j,a} &= \{b \in B_{p,q} \mid b_j = 1 \text{ and } b_0 = a\} \\
C_{i,j} &= \{b \in B_{p,q} \mid b_i = b_j = 1\} \\
C_i &= \{b \in B_{p,q} \mid b_i = 1\}
\end{aligned}$$

Then we have, by setting $B = B_{p,q}$

$$\begin{aligned}
S(\mathbf{a}(x_i)) &= K \cap B^* C_{i,a} B^\infty \\
S(x_i = x_j) &= K \cap B^* C_{i,j} B^\infty \\
S(x_i < x_j) &= K \cap B^* C_i B^* C_j B^\infty \\
S(X_i(x_j)) &= K \cap B^* C_{i+p,j} B^\infty
\end{aligned}$$

which establishes the proposition. □

Proposition 3.4 allows one to treat logical connectives. It remains to treat the case of the formulas of the form $\exists x \varphi$ and $\exists X \varphi$. Denote by π_i the function consisting to "erase" the i-th component, defined by

$$\pi_i(b_0, b_1, \ldots, b_{p+q}) = (b_0, b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_{p+q})$$

Thus π_i should be considered as a function from $B_{p,q}$ into $B_{p-1,q}$ if $i \leq p$ and into $B_{p,q-1}$ if $p < i \leq p+q$.

Proposition 3.6 *For each formula φ, the following formulas hold*

(1) $S_{p-1,q}(\exists x_p \varphi) = \pi_p(S_{p,q}(\varphi))$
(2) $S_{p,q-1}(\exists X_q \varphi) = \pi_{p+q}(S_{p,q}(\varphi))$

Proof. This follows from the definition of existential quantifiers. □

We are now ready to show that if $S(\varphi)$ is rational, then so are $S(\exists x \varphi)$ and $S(\exists X \varphi)$. We may assume that $x = x_p$ and $X = X_q$. Then, by Proposition 3.6, we have $S(\exists x_p \varphi) = \pi_p(S(\varphi))$ and $S(\exists X_q \varphi) = \pi_{p+q}(S(\varphi))$. Since morphisms preserve rationality, the result follows.

This concludes the proof of Büchi's theorem. The reader will find in Exercise 4 the sketch of another proof to pass from words to formulas based on a traduction of rational expressions.

We now show, as it was announced in Section 2.3 that strong and weak logic coincide on words.

Theorem 3.7 *The theories $MF_2(<)$ and $WMF_2(<)$ are equivalent on words.*

Proof. Proposition 2.7 shows that $WMF_2(<) \leq MF_2(<)$. To establish the opposite direction, it suffices, by Büchi's theorem, to show that each rational subset of A^∞ can be defined by a formula of $WMF_2(<)$. There is no problem for a set of finite words, since set variables are necessarily interpreted as finite sets. Therefore it suffices to consider the case of a recognizable set X of infinite words.

By McNaughton's theorem (Theorem I.9.1), X is a boolean combination of sets of the form $\overrightarrow{Y}$. We claim that such a set is definable in weak monadic second order logic. Indeed, let $\varphi(x, y)$ be a formula of $WMF_2(<)$ defining Y (which consists of finite words). Then we have $u \models \varphi(0, n)$ if and only if $u_0 u_1 \ldots u_{n-1} \in Y$ by Proposition 2.6 and the formula

$$\psi = \forall x\bigl(\exists y(x < y) \wedge \varphi(0, y)\bigr)$$

guaranties that u has infinitely many prefixes in Y. Thus $S(\psi) = \overrightarrow{Y}$. □

Another proof of this result is given in Exercise 5. It consists in coding directly a Muller automaton by a weak monadic second order formula. We conclude this section by making two observations of logical nature.

First of all, Büchi's theorem allows one to show that the monadic second order theory of non negative integers is decidable.

Theorem 3.8 *There is an algorithm to decide whether a given statement of $MF_2(<)$ is true on $\mathbb{N}$.*

Proof. Let $\varphi \in MF_2(<)$. One can effectively calculate the set $S(\varphi)$ and decide whether $S(\varphi) = A^\omega$ or $S(\varphi) = \emptyset$. □

Next, the proof of Proposition 3.2 shows that the hierarchy on formulas of $MF_2(<)$ based on the number of alternations of second order quantifiers collapses to the first level. We have indeed the following result.

Proposition 3.9 *Every formula of $MF_2(<)$ is equivalent to a formula of the form*

$$\exists X_1 \ldots \exists X_k \varphi(X_1, \ldots, X_k)$$

where φ is a first order formula.

In fact, one can even show that every formula of $MF_2(<)$ is equivalent to a formula of the form $\exists X \varphi(X)$ where φ is a first order formula (Exercise 2).

4 First order logic of the linear order.

We now study the language $F_1(<)$ of the first order logic of the linear order. We shall, in a first step, characterize the sets of words definable in this logic, which happen to be the star-free sets. Next, we shall see in Section 4.2 how this result can be refined to establish a correspondence between the levels the Σ_n-hierarchy of first order logic and the concatenation hierarchy of star-free sets.

4.1 First order and star-free sets.

We shall prove the following key result.

Theorem 4.1 *A set of finite or infinite words is star-free if and only if it is definable by a formula of $F_1(<)$.*