

Semantics and Verifications

Based on the lectures of Colin RIBA
Notes written by Hugo SALOU



ENS DE LYON

December 14, 2025
printed version

Contents

1	Introduction.	4
2	Transition systems.	5
2.1	Transition systems.	5
2.2	Program graphs.	6
2.3	Transition system of a program graph.	7
3	Linear Time Properties.	8
3.1	Linear-time properties.	8
3.2	Decomposition of a linear-time property.	11
3.2.1	Safety properties.	11
3.2.2	Safety properties and trace equivalences.	12
3.2.3	Propriétés de vivacité.	17
4	L'approche topologique.	18
4.1	Théorème de décomposition.	19
4.2	Bases.	20
5	Ordres partiels et treillis.	21
5.1	Ordres partiels.	21
5.2	Treillis complet.	21
5.3	Opérateur de clôture.	23
5.4	Connexion de Galois.	23
6	Propriétés “observables”.	26
6.1	Compacité.	28
6.2	Espace Hausdorff.	29
7	Logique temporelle linéaire.	29
7.1	La logique LML.	29
7.1.1	Équivalences logiques.	31
7.1.2	Homework : Dualité de Stone.	31
7.1.3	Extension de LML avec \square et \Diamond	32
7.1.4	Théorème de Knaster-Tarski.	34
7.2	La logique LTL.	35
7.2.1	Points fixes dans LTL.	35
7.2.2	Équivalences logiques pour LTL.	36
8	Langages ω-réguliers.	37
8.1	Expressions ω -régulières.	37
8.2	Automates de Büchi non-déterministes (NBA).	38
8.3	Automates de Büchi déterministes (DBA).	41
8.4	Automates de Büchi généralisés (GNBA).	42

8.5 Traduction de LTL en (G)NBA.	43
8.6 Théorème de Büchi : complémentation pour les langages ω -réguliers.	45
9 Bisimulation.	48
9.1 Quotient par bisimulation.	50
9.2 Bisimilarité et équivalence de traces.	50
10 Logique modale pour systèmes de transitions.	51
10.1 Modèle de Kripke.	51
10.2 Logique de Hennessy-Milner.	51
10.3 Équivalences logiques.	53
10.3.1 Équivalences logiques pour les formules.	53
10.3.2 Équivalences logiques pour les états.	53
10.4 Propriété de Hennessy-Milner.	54
10.5 Saturation modale.	54
10.6 Algèbres de Boole avec opérateurs.	55
10.7 Structures ultrafiltres.	57
10.7.1 Extensions d'ultrafiltres aux modèles de Kripke.	57

I. Introduction.

Let us give precisions on the terms in the name of the course, and in the broader space of semantics and verifications.

Verification. Formal techniques to ensure the correctness software or hardware of systems.

Model Checking. “Automatic” checking of the correctness by means of exhaustive exploration.

Exemple 1.1. Consider a program that is 10 lines long, contains 3 booleans variables and 5 integers variables in the range $\{0, \dots, 9\}$. The number of states for this program is:

$$10 \times 2^3 \times 10^5 = 8\,000\,000.$$

The real issue with the state exploration problem is the factor 10^5 , coming from the use of 5 integers.

Exemple 1.2. Consider a server and n clients. Clients can make requests to the server and the server can answer a client. The specification of this server should include the following:

- ▷ Each client which makes a request is eventually answered.
- ▷ We *abstract* away from precise quantitative constraints.

We will sometimes reason about an infinite amount of executions. For example, if some client makes infinitely-many requests (then it'll have infinitely-many answers). Infinite sequences are represented by ω -words, *i.e.* infinite words indexed by \mathbb{N} . Thus, ω -words on some alphabet Σ are functions $\mathbb{N} \rightarrow \Sigma$. We will denote Σ^ω the set of those infinite words on the alphabet Σ .

If $|\Sigma| \geq 2$, then the set Σ^ω is uncountable.

This course will cover the following:

- ▷ Transition systems;
- ▷ Linear-time properties;
- ▷ Topology;
- ▷ Orders and Lattices;
- ▷ Linear Temporal Logic (LTL);
- ▷ Büchi automata;
- ▷ **Stone duality** (mostly in homework);
- ▷ Bisimilarity/bisimulation;
- ▷ Modal Logic.

Ressources from this course include:

- ▷ the [course notes](#) (available online, non-exhaustive);
- ▷ Baier, C. and Katoen, J.-P., Principles of Model Checking, MIT Press, 2008.

Prerequisites for this course include:

- ▷ First-order logic (see my [course notes](#) for the “Logique” L3 course, in french);
- ▷ Finite automata (“FDI” L3 course).

Evaluation for this course will be in two parts: the final exam (50 %) and the homework, in two parts (25 % each).

The tutorials will be done by Lison Blondeau-Patissier.

II. Transition systems.

2.1 Transition systems.

Définition 2.1. A transition system is a tuple

$$TS = (S, \text{Act}, \rightarrow, I, \text{AP}, L)$$

where

- ▷ S is the set of *states*;
- ▷ Act is the set of *actions*;
- ▷ $\rightarrow \subseteq S \times \text{Act} \times S$ the *transition relation*;
- ▷ $I \subseteq S$ the set of *initial states*;
- ▷ AP is the set of *atomic propositions*;
- ▷ $L : S \rightarrow \wp(\text{AP}) \cong 2^{\text{AP}}$ is the *state labelling function*.

We will write $s \xrightarrow{\alpha} s'$ when $(s, \alpha, s') \in \rightarrow$.

Exemple 2.1 (Beverage Vending Machine, BVM). We can model a beverage vending machine using a diagram like in figure 2.1. Here we have that:

- ▷ $S = \{\text{pay}, \text{select}, \text{soda}, \text{beer}\}$,
- ▷ $I = \{\text{pay}\}$,
- ▷ $\text{Act} = \{\text{ic}, \tau, \text{gb}, \text{gs}\}$.¹

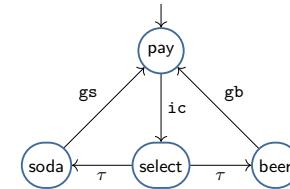


Figure 2.1 | Transition system for the BVM

We can define the labels:

$$L(\text{pay}) = \emptyset \quad L(\text{soda}) = L(\text{beer}) = \{\text{paid, drink}\} \quad L(\text{select}) = \{\text{paid}\},$$

with $\text{AP} = \{\text{paid, drink}\}$.

2.2 Program graphs.

The goal is to represent the evaluation of a program.

Définition 2.2 (Typed variables). A set Var of *variables*.

- ▷ For each variable $x \in \text{Var}$, consider a set $\text{Dom}(x)$.
- ▷ Given $\text{TV} = (\text{Var}, (\text{Dom}(x))_{x \in \text{Var}})$, we define

$$\text{Eval}(\text{TV}) = \prod_{x \in \text{Var}} \text{Dom}(x),$$

the set of valuations of the form $\eta : x \in \text{Var} \mapsto \eta(x) \in \text{Dom}(x)$ (in the sense of a dependent function type).

Définition 2.3 (Program graph). A *program graph* is a tuple

$$PG = (\text{Loc}, \text{Act}, \text{Effect}, \hookrightarrow, \text{Loc}_0, g_0),$$

where

- ▷ Loc is the set of *locations* (lines of codes);
- ▷ Act is the set of *actions*;
- ▷ $\text{Effect} : \text{Act} \times \text{Eval}(\text{TV}) \rightarrow \text{Eval}(\text{TV})$;
- ▷ $\hookrightarrow \subseteq \text{Loc} \times \text{Conditions} \times \text{Act} \times \text{Loc}$ where conditions are propositional formula built from atoms of the forms “ $x \in D$ ” for some variable x and some set $D \subseteq \text{Dom}(x)$;
- ▷ $\text{Loc}_0 \subseteq \text{Loc}$ the set of *initial locations*;
- ▷ g_0 is the *initial condition*.

We will write $\ell \xrightarrow{g:\alpha} \ell'$ for $(\ell, g, \alpha, \ell') \in \hookrightarrow$.

¹The meaning of the actions are the following: ic means *insert coin*, gb means *get beer* and gs for *get soda*.

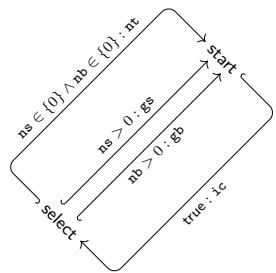


Figure 2.2 | BVM as a program graph

Exemple 2.2 (BVM as a program graph). In figure 2.2, we use

- ▷ $\text{Loc} = \{\text{start}, \text{select}\};$
- ▷ $\text{Var} = \{\text{ns}, \text{nb}\};$
- ▷ $\text{Act} = \{\text{ic}, \text{nt}, \text{gs}, \text{gb}, \text{refill}\};$
- ▷ $\text{Loc}_0 = \{\text{start}\};$
- ▷ $g_0 = \text{ns} \in \{\text{max}\} \wedge \text{nb} \in \{\text{max}\}$
- ▷

$$\begin{aligned} \text{Effect} : \text{Act} \times \text{Eval}(TV) &\longrightarrow \text{Eval}(TV) \\ (\text{refill}, \eta) &\longmapsto [\text{ns} \mapsto \text{max}, \text{nb} \mapsto \text{max}] \\ (\text{gs}, \eta) &\longmapsto \eta[\text{ns} \mapsto \eta(\text{ns}) - 1] \\ (\text{gb}, \eta) &\longmapsto \eta[\text{nb} \mapsto \eta(\text{nb}) - 1] \end{aligned}$$

2.3 Transition system of a program graph.

Définition 2.4. Given TV and PG a program graph, we define

$$TS(PG) := (S, \text{Act}, \rightarrow, I, \text{AP}, L)$$

where

- ▷ $S = \text{Loc} \times \text{Eval}(TV);$
- ▷ $\text{AP} = \text{Loc} \cup \text{Conditions} ;$
- ▷ $I = \{(\ell_0, \eta) \mid \ell_0 \in \text{Loc}_0, \eta \models g_0\};$
- ▷ \rightarrow is defined by:

$$\frac{\ell \xrightarrow{g:\alpha} \ell' \quad \eta \models g}{(\ell, \eta) \xrightarrow{\alpha} (\ell', \text{Effect}(\alpha, \eta))},$$

- ▷ and $L(\ell, \eta) = \{\ell\} \cup \{g \mid \eta \models g\}.$

Exemple 2.3. The BVM program graph example seen in the previous example can be transformed as a transition system thanks to the previous definition; it is shown in figure 2.3. To simplify, we assume $\text{max} = 1$.

III. Linear Time Properties.

Definition 3.1. Let Σ be an alphabet (i.e. a set).

1. A ω -word on Σ is a function $\sigma : \mathbb{N} \rightarrow \Sigma$. We denote Σ^ω for the set of ω -words on Σ .
2. We define $\Sigma^\infty := \Sigma^\omega \cup \Sigma^*$ the set of finite or infinite words.
3. Given $\hat{\sigma} \in \Sigma^*$ and $\sigma \in \Sigma^\infty$, we say that $\hat{\sigma}$ is a prefix of σ , written $\hat{\sigma} \subseteq \sigma$, whenever $\forall i < \text{length}(\hat{\sigma}), \hat{\sigma}(i) = \sigma(i)$.
4. Given $\sigma \in \Sigma^\infty$, we define

$$\text{Pref}(\sigma) := \{\hat{\sigma} \in \Sigma^* \mid \hat{\sigma} \subseteq \sigma\},$$

which we extend to sets of words: for $E \subseteq \Sigma^\infty$,

$$\text{Pref}(E) := \bigcup_{\sigma \in E} \text{Pref}(\sigma).$$

Remark 3.1. ▷ The prefix order \subseteq on Σ^* is generally¹ a partial order: there are $u, v \in \Sigma^*$ such that $u \not\subseteq v$ and $v \not\subseteq u$.

▷ Given $\sigma \in \Sigma^\infty$, the prefix order \subseteq on $\text{Prefix}(\sigma)$ is a linear (or total order).

3.1 Linear-time properties.

Let AP be a set of atomic propositions.

Definition 3.2. A linear-time property (sometimes written LT property) on AP is a set $P \subseteq (\mathbf{2}^{\text{AP}})^\omega$.

The idea is that a linear-time property $A : \mathbb{N} \rightarrow \mathbf{2}^{\text{AP}}$ specifies, for each $i \in \mathbb{N}$, a set $\sigma(i) \subseteq \text{AP}$ of all atomic propositions assumed at time i .

Example 3.1. For the Beverage vending machine (shown in figure 3.1), we can have the following linear-time properties:

¹As long as the alphabet has at least two letters.

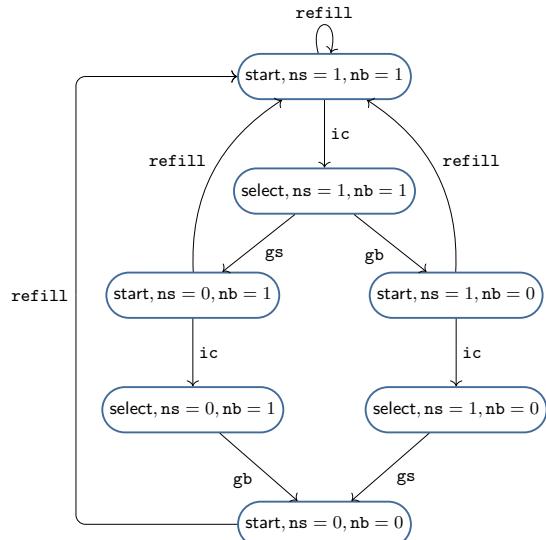


Figure 2.3 | Transition system of the BVM program graph

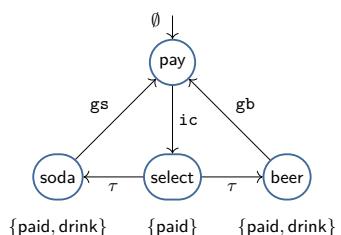


Figure 3.1 | Transition system for the BVM with labels

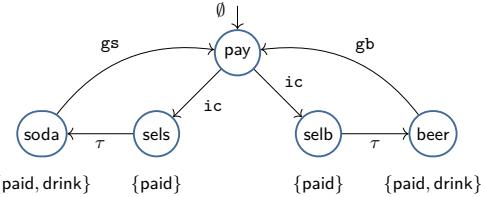


Figure 3.2 | Transition system for the alternative BVM

- ▷ $\{\sigma \in (\mathbf{2}^{\text{AP}})^\omega \mid \forall n \in \mathbb{N}, \text{drink} \in \sigma(n) \implies \exists k < n, \text{paid} \in \sigma(k)\}$,
- ▷ $\{\sigma \in (\mathbf{2}^{\text{AP}})^\omega \mid \forall n \in \mathbb{N}, \#\{k \leq n \mid \text{drink} \in \sigma(k)\} \leq \#\{k \leq n \mid \text{paid} \in \sigma(k)\}\}$,
- ▷ $\{\sigma \in (\mathbf{2}^{\text{AP}})^\omega \mid (\exists^\infty t, \text{paid} \in \sigma(t)) \implies (\exists^\infty t, \text{drink} \in \sigma(t))\}$,
- ▷ $\{\sigma \in (\mathbf{2}^{\text{AP}})^\omega \mid (\forall^\infty t, \text{paid} \notin \sigma(t)) \implies (\forall^\infty t, \text{drink} \notin \sigma(t))\}$.

Remark 3.2. The notations \exists^∞ and \forall^∞ are “infinitely many” and “ultimately all” quantifiers:

- ▷ $\exists^\infty t, P(t)$ is, by definition, $\forall N \in \mathbb{N}, \exists t \geq N, P(t)$;
- ▷ $\forall^\infty t, P(t)$ is, by definition, $\exists N \in \mathbb{N}, \forall t \geq N, P(t)$.

Definition 3.3. A (finite or infinite) *path* in *TS* is a finite or infinite sequence $\pi = (s_i)_i \in S^\infty$ which respects transitions: for all i , we have $s_i \xrightarrow{a} s_{i+1}$ for some $a \in \text{Act}$. A path $\pi = (s_i)_i$ is *initial* if $s_0 \in I$.

Definition 3.4 (Trace). 1. The *trace* of a path $\pi = (s_i)_i$ is the (finite or infinite) word

$$L(\pi) := (L(s_i))_i \in L^\infty.$$

2. We define

- ▷ $\text{Tr}(TS) := \{L(\pi) \mid \pi \text{ is a finite or infinite path in } TS\}$;
- ▷ $\text{Tr}^\omega(TS) := \{L(\pi) \mid \pi \text{ is an infinite path in } TS\}$;
- ▷ $\text{Tr}_{\text{fin}}(TS) := \{L(\pi) \mid \pi \text{ is a finite path in } TS\}$.

Definition 3.5 (Satisfaction of a LT property). We say that a transition system *TS* over AP *satisfies* a LT property *P* on AP, written $TS \approx P$, when $\text{Tr}^\omega(TS) \subseteq P$.

Example 3.2. The BVM satisfies all the properties from example 3.1.

Example 3.3. We use a different transition system *BVM'* to model the beverage vending machine, as seen in figure 3.2. The two transition systems are equivalent in the sense that:

$$\text{Tr}^\omega(\text{BVM}') = \text{Tr}^\omega(\text{BVM}),$$

so, for any LT Property $P \subseteq (\mathbf{2}^{\text{AP}})^\omega$,

$$\text{BVM}' \approx P \quad \text{iff} \quad \text{BVM} \approx P.$$

We have a very simple result, which we will (probably) prove in the tutorials.

Proposition 3.1. Given two transition systems TS_1 and TS_2 over AP, then the following are equivalent:

- ▷ $\text{Tr}^\omega(TS_1) \subseteq \text{Tr}^\omega(TS_2)$,
- ▷ $\forall P \subseteq (2^{\text{AP}})^\omega, TS_2 \approx P \implies TS_1 \approx P$.

3.2 Decomposition of a linear-time property.

In this section, we introduce the notions of a “safety property” and a “liveness property” such that, for any LT property P ,

1. there exists a safety property P_{safe} and a liveness property P_{liveness} such that

$$P = P_{\text{safe}} \cap P_{\text{liveness}};$$

2. P is a liveness and a safety property if and only if $P = (2^{\text{AP}})^\omega$.

3.2.1 Safety properties.

The idea of a safety property is to ensure that “nothing bad is going to happen.”

Definition 3.6. We say that $P \subseteq (2^{\text{AP}})^\omega$ is a *safety property* if there exists a set $P_{\text{bad}} \subseteq (2^{\text{AP}})^*$ such that

$$\sigma \in P \iff \text{Pref}(\sigma) \cap P_{\text{bad}} = \emptyset.$$

Example 3.4. Considering the examples of LT-properties from example 3.1,

- ▷ Property (1) is a safety property: we can consider

$$P_{\text{bad}}^{(1)} = \{\hat{\sigma} \in \Sigma^* \mid \text{drink} \in \hat{\sigma}(n) \wedge \forall i < n, \text{paid} \notin \hat{\sigma}(i)\},$$

where n is the length of $\hat{\sigma}$.

- ▷ Property (2) is a safety property: we can consider

$$P_{\text{bad}}^{(2)} = \{\hat{\sigma} \in \Sigma^* \mid \#\{t \mid \text{paid} \in \hat{\sigma}(t)\} < \#\{t \mid \text{drink} \in \hat{\sigma}(t)\}\}.$$

- ▷ Properties (3) and (4) are not safety properties: for any finite word $\hat{\sigma} \in (2^{\text{AP}})^\omega$, there exists $\sigma \in (2^{\text{AP}})^\omega$ such that $\hat{\sigma} \subseteq \sigma$ and $\sigma \in P$.

Example 3.5 (Traffic Light). We consider a traffic light as a transition system over $\text{AP} = \{\text{G}, \text{Y}, \text{R}\}$, as shown in figure 3.3. An example of a safety property is

$$\forall n, \text{R} \in \sigma(n) \implies n > 0 \text{ and } \text{Y} \in \sigma(n-1).$$

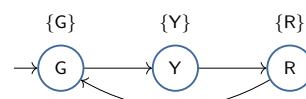


Figure 3.3 | Transition system for the traffic light



Figure 3.4 | Another transition system



Figure 3.5 | One-state transition system

3.2.2 Safety properties and trace equivalences.

Example 3.6. Consider the transition system shown in figure 3.4, a safety property P with $P_{\text{bad}} = \{a\}^* \{b\}$ is satisfied: $TS \approx P$. This is true since $\text{Tr}^\omega(TS) = \{a\}^\omega$. However, when we consider *finite* (instead of infinites) traces, we have that $\text{Tr}_{\text{fin}}(TS) \cap P_{\text{bad}} \neq \emptyset$.

Definition 3.7 (Terminal state). A state $s \in S$ of a transition system TS is *terminal* if

$$\forall s' \in S, \forall \alpha \in \text{Act}, s \xrightarrow{\alpha} s'.$$

Proposition 3.2. Let TS be a transition system without terminal states, and a safety property P with P_{bad} the set of “bad behaviours”. Then,

$$TS \approx P \quad \text{if and only if} \quad \text{Tr}_{\text{fin}}(TS) \cap P_{\text{bad}} = \emptyset.$$

Proof. See the course notes in § 3.2.3. \square

Lemma 3.1. Let TS and TS' be two transition systems over AP without terminal states. Then, the following are equivalent:

- ▷ $\text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS')$;
- ▷ for any safety property P , $TS' \approx P$ implies $TS \approx P$.

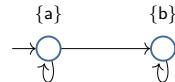
Proof. $\triangleright \Rightarrow$. This is true by the last proposition.

- ▷ \Leftarrow . Let P be a safety property with

$$P_{\text{bad}} = (2^{\text{AP}})^* \setminus \text{Tr}_{\text{fin}}(TS').$$

So, $TS' \not\approx P$ hence $TS \not\approx P$ by assumption. Therefore, $\text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS')$ by the last proposition. \square

Example 3.7. Consider the transition system TS from figure 3.4 (example 3.6, which has a terminal state), and the transition system TS' from figure 3.5. Safety properties satisfied in TS' are satisfied in TS . However, $\text{Tr}_{\text{fin}}(TS) \not\subseteq \text{Tr}_{\text{fin}}(TS')$ (even though the sets of infinite traces are equal).

**Figure 3.6** | A transition system with no terminal state

Example 3.8. Consider the transition system TS' shown in figure 3.7 (page 14) and TS the transition system shown in figure 3.6. The transition system TS' has terminal states and TS does not. However, we have that

$$\text{Tr}_{\text{fin}}(TS) = \text{Tr}_{\text{fin}}(TS').$$

We also have that

$$\text{Tr}^\omega(TS') = \{a\}^\omega \quad \text{and} \quad \text{Tr}^\omega(TS) = \{a\}^\omega \cup \{a\}^+ \cdot \{b\}^\omega.$$

Thus giving a counter example to the previous lemma if one of the transition system has terminal states: consider a linear-time property P with the set of “bad behaviors” as $P_{\text{bad}} = \{a\}^+ \cdot \{b\}$, then $TS' \not\models P$ but $TS \models P$.

The rest of the course will be done in french.

L’objectif est de trouver des conditions sur TS et TS' telles que l’on ait l’équivalence entre :

- ▷ $\text{Tr}^\omega(TS) \subseteq \text{Tr}^\omega(TS')$;
- ▷ pour toute propriété de sûreté P , $TS' \not\models P$ implique $TS \not\models P$.

On commence par trouver des conditions sur TS et TS' telles que l’on ait l’équivalence :

$$\text{Tr}^\omega(TS) \subseteq \text{Tr}^\omega(TS') \iff \text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS').$$

Pour que l’on ait « \iff », il est nécessaire que TS soit sans état terminal. En effet, si TS est sans état terminal, alors pour tout $\hat{\sigma} \in \text{Tr}_{\text{fin}}(TS)$, il existe $\sigma \in \text{Tr}^\omega(TS)$ tel que $\hat{\sigma} \subseteq \sigma$.

Dans l’autre sens, supposons que $\text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS')$. Soit $\sigma \in \text{Tr}^\omega(TS)$. Alors, pour tout $\hat{\sigma} \subseteq \sigma$, on a $\hat{\sigma} \in \text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS')$. Ainsi, pour tout $n \in \mathbb{N}$, on a qu’il existe un chemin $\pi^n := (\pi_i^n)_{i \leq n}$ initial dans TS' tel que $L(\pi^n) = \sigma(0) \dots \sigma(n)$.

Exemple 3.1. On considère TS comme celui représenté en figure 3.6 et TS' comme représenté en figure 3.8. On a que

$$\text{Tr}_{\text{fin}}(TS) = \text{Tr}_{\text{fin}}(TS') = \{a\}^* \cup \{a\}^+ \cdot \{b\}^*,$$

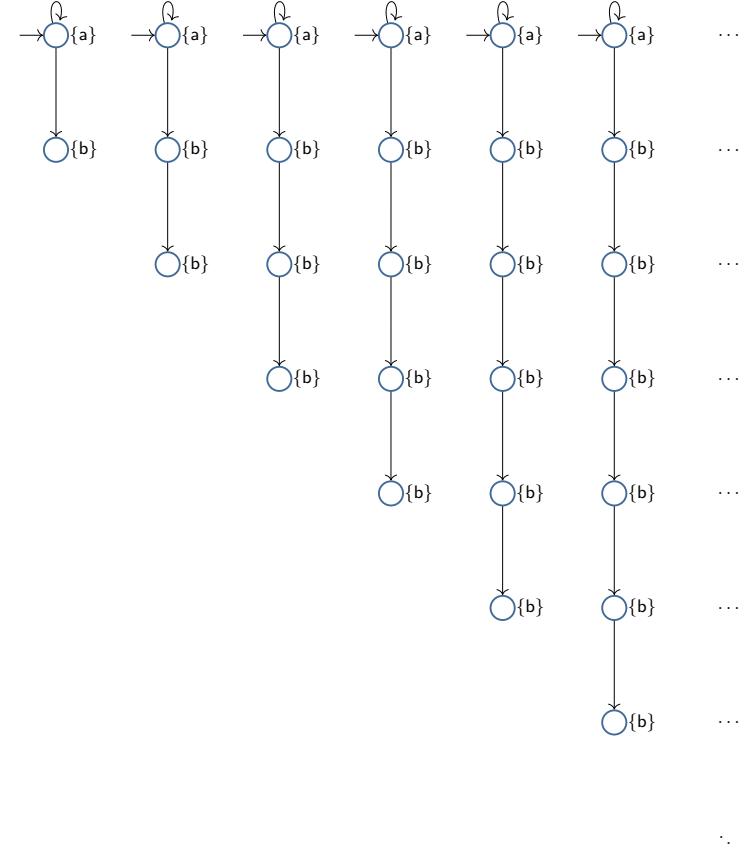
et

$$\text{Tr}^\omega(TS) = \{a\}^\omega \cup \{a\}^+ \cdot \{b\}^\omega \neq \{a\}^+ \cdot \{b\}^\omega = \text{Tr}^\omega(TS').$$

Dans notre cas, on a $\{a\}^n \subseteq \text{Tr}_{\text{fin}}(TS')$ mais on n’a pas $\{a\}^\omega \subseteq \text{Tr}^\omega(TS')$.

Définition 3.1 (Branchement fini). Un système de transition $TS = (S, \text{Act}, \rightarrow, I, \text{AP}, L)$ est à branchement fini si

1. I est fini ;
2. pour tout $s \in S$, l’ensemble $\{s' \mid \exists \alpha \in \text{Act}, s \xrightarrow{\alpha} s'\}$ est fini.



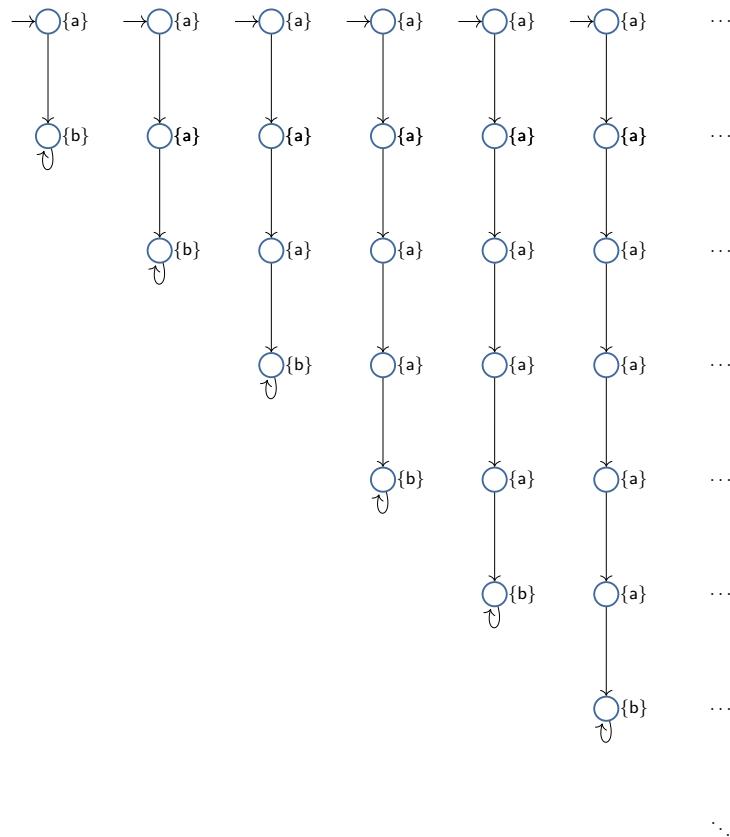
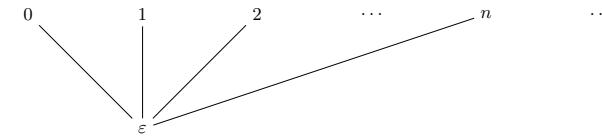


Fig. 3.8 | Un système de transition infini

Fig. 3.9 | Arbre $\{\varepsilon\} \cup \mathbb{N}$ sur \mathbb{N} **Interlude. Le lemme de König.****Définition 3.2.** Soit A un ensemble.

1. Un arbre sur A est un ensemble $T \subseteq A^*$ clos par préfixe, c'est-à-dire que si $u \in T$ alors pour tout préfixe $v \subseteq u$, on a $v \in T$.
2. Un chemin infini dans un arbre $T \subseteq A^*$ est un mot $\pi \in A^\omega$ tel que, pour tout $n \in \mathbb{N}$, $\pi(0) \dots \pi(n) \in T$.
3. Un arbre est à branchement fini si, pour tout $u \in T$, l'ensemble $\{ua \mid a \in A \text{ et } ua \in T\}$ est fini.

Remarque 3.1. Si A est fini, alors tout arbre sur A est à branchement fini. Aussi, si T est fini alors T n'a pas de chemin infini.**Exemple 3.2.** Avec $T \subseteq \mathbb{N}^*$ défini par $T = \{\varepsilon\} \cup \mathbb{N}$ alors T est sans chemin infini et à branchement infini (figure 3.9).**Lemme 3.1 (Lemme de König).** Si T est un arbre infini et à branchement fini alors T a un chemin infini.**Preuve.** Soit $T \subseteq A^*$ un arbre infini à branchement fini. Si $u \in T$ on note $T \upharpoonright u$ l'arbre

$$\{v \in T \mid u \subseteq v \text{ ou } v \subseteq u\}.$$

On remarque que $T = T \upharpoonright \varepsilon$ et $T \upharpoonright u = \bigcup_{a \in A} ua \in T} T \upharpoonright ua$. Alors, comme que T est infini et $T = \bigcup_{a \in A \cap T} T \upharpoonright a$, par le lemme des tiroirs infinis, il existe $a \in A \cap T$ tel que $T \upharpoonright a$ est infini. On a donc

$$T \upharpoonright a = \bigcup_{b \in A} ab \in T.$$

Par induction sur $n \in \mathbb{N}$, on définit $a_0, \dots, a_n \in A$ (en étendant) tel que $a_0 \dots a_n \in T$ et $T \upharpoonright a_0 \dots a_n$ est infini. On obtient donc $\pi = (a_i)_{i \in \mathbb{N}}$ qui est un chemin infini dans T . \square

Remarque 3.2 (Attention!). On doit manipuler un arbre !En considérant $A = \{0, 1\}$ avec $T_0 = \{0\}^* \{1\} \subseteq A^*$, on a que :

- ▷ T_0 est infini;
- ▷ T_0 est à branchement fini;
- ▷ MAIS, ce n'est pas un arbre.

On considère donc $T = \text{Pref}(T_0)$ qui est un arbre infini et à branchement fini. Alors, par le lemme de König, on a que T a un chemin infini $\pi \in \{0\}^\omega$ (il n'y a qu'un seul choix possible). Et, on a $\text{Pref}(\pi) \subseteq T$ sauf que $\text{Pref}(\pi) \cap T_0 = \emptyset$.

**Fig. 3.10 |** Un feu tricolore un peu dangereux

On peut maintenant revenir à notre objectif de caractériser les propriétés de sûreté par les traces.

Proposition 3.1. Si TS est sans état terminal et TS' est à branchement fini, on a que

$$\text{Tr}^\omega(TS) \subseteq \text{Tr}^\omega(TS') \iff \text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS').$$

Preuve. ▷ « \implies ». On l'a déjà vu précédemment.

▷ « \iff ». Supposons $\text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS')$. Considérons un mot $\sigma \in \text{Tr}^\omega(TS)$. Soit $T' \subseteq (S')^*$ (où S' est l'ensemble des états de TS') défini par

$$T' = \{u \in (S')^* \mid u \text{ chemin initial fini de } TS' \text{ et } L'(u) \subseteq \sigma\}.$$

On a que T' est un arbre, qui est infini (car $\text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS')$). Aussi, on a que T' est à branchement fini car TS' est à branchement fini. Par le lemme de König, on a que T' a un chemin infini π . On a aussi $L'(\pi) = \sigma$ et donc $\sigma \in \text{Tr}^\omega(TS')$. \square

Corollaire 3.1. Si TS et TS' sont deux systèmes de transitions sans états terminaux et à branchement fini alors les deux propriétés suivantes sont équivalentes :

- ▷ $\text{Tr}^\omega(TS) = \text{Tr}^\omega(TS')$;
- ▷ pour toute propriété de sûreté P , $TS' \approx P$ ssi $TS \approx P$. \square

3.2.3 Propriétés de vivacité.

L'idée est de s'assurer que « quelque chose de bon peut toujours arriver ».

Exemple 3.3. Avec $AP = \{G, Y, R\}$ et TS défini comme en figure 3.10. On a que TS satisfait « si R à un instant donné alors Y à l'instant précédent » (on note cette propriété P_{safe}). Cependant TS ne satisfait pas $P_{\text{live}} := \{\sigma \mid \exists^\infty t, R \in \sigma(t)\}$.

Définition 3.3 (Vivacité). On dit que $P \subseteq (\mathbf{2}^{AP})^\omega$ est une propriété de *vivacité* si, pour tout mot fini $\hat{\sigma} \in (\mathbf{2}^{AP})^*$, il existe $\sigma \in (\mathbf{2}^{AP})^\omega$ tel que $\hat{\sigma} \subseteq \sigma$ et $\sigma \in P$.

Exemple 3.4. Avec l'exemple de la BVM, les propriétés (3) et (4) sont des propriétés de vivacité.

Dans la suite, on montrera le théorème de décomposition suivant en passant au point de vue topologique.

Théorème 3.1. Pour toute propriété $P \subseteq (\mathbf{2}^{AP})^\omega$, il existe

- ▷ P_{safe} une propriété de sûreté,
- ▷ P_{live} une propriété de vivacité,

tels que $P = P_{\text{safe}} \cap P_{\text{live}}$.

Proposition 3.2. La propriété $\text{True} := (\mathbf{2}^{AP})^\omega \subseteq (\mathbf{2}^{AP})^\omega$ est l'unique LT-property sur AP qui est une propriété de sûreté et de vivacité.

Preuve. ▷ On a que True est une propriété de sûreté en posant l'ensemble des « mauvais comportements » comme $\text{True}_{\text{bad}} := \emptyset$.

▷ On a que True est une propriété de vivacité car, pour tout mot fini $\hat{\sigma} \in (\mathbf{2}^{AP})^*$, il existe $\sigma \in (\mathbf{2}^{AP})^\omega$ tel que $\hat{\sigma} \subseteq \sigma$.

▷ **Unicité.** Soit $P \subseteq (\mathbf{2}^{AP})^\omega$ de sûreté pour $P_{\text{bad}} \subseteq (\mathbf{2}^{AP})^*$. Si P est une propriété de vivacité alors pour tout $\hat{\sigma} \in P_{\text{bad}}$ alors il existe $\sigma \in P$ tel que $\hat{\sigma} \subseteq \sigma$. Donc, on a que $P_{\text{bad}} = \emptyset$ et donc $P = \text{True}$. \square

IV.L'approche topologique.

Définition 4.1. Un *espace topologique* est une paire $(X, \Omega X)$ où X est un ensemble et $\Omega X \subseteq \wp(X)$ que l'on appelle *ensemble des ouverts* telle que

- ▷ si $\mathcal{S} \subseteq_{\text{fin}} \Omega X$ alors $\bigcap \mathcal{S} = \bigcap_{V \in \mathcal{S}} V \in \Omega X$;
- ▷ si $\mathcal{S} \subseteq \Omega X$ alors $\bigcup \mathcal{S} = \bigcup_{V \in \mathcal{S}} V \in \Omega X$.

Remarque 4.1. On a toujours $\emptyset, X \in \Omega X$ avec $\emptyset = \bigcup \emptyset$ et $X = \bigcap \emptyset$.

Exemple 4.1 (Topologie sur Σ^ω et intuition). On peut voir les ouverts comme « analogues » aux ensembles récursivement énumérables.

On définit une topologie sur Σ^ω où les ouverts sont $\text{ext}(W)$ où $W \subseteq \Sigma^*$ et $\text{ext}(W) = \bigcup_{u \in W} \text{ext}(u)$ et

$$\text{ext}(u) = \{\sigma \in \Sigma^\omega \mid u \subseteq \sigma\}.$$

Ainsi, si on a une manière d'énumérer W , on peut tester si $u \in \text{ext}(W)$ en temps fini, mais il n'est pas forcément possible de vérifier que $u \notin \text{ext}(W)$.

Définition 4.2. Soit $(X, \Omega X)$ un espace topologique. Alors, on appelle *fermé* un sous-ensemble $C \subseteq X$ tel que $X \setminus C \in \Omega X$.

Remarque 4.2. On a donc que \emptyset et X sont toujours fermés.

Remarque 4.3. L'ensemble des fermés sur $(X, \Omega X)$ est stable par

- ▷ intersections arbitraires ;
- ▷ unions finies.

Ce sont les « *duales* » des propriétés de stabilité des ouverts.

Avec quelques manipulations « simples », on peut arriver à la caractérisation suivante.

Lemme 4.1. Soit $(X, \Omega X)$ un espace topologique.

▷ On a que $A \subseteq X$ est un ouvert ssi $\forall x \in X$ on a l'équivalence suivante

$$x \in A \iff \exists U \in \Omega X, \quad x \in U \subseteq A.$$

▷ On a que $A \subseteq X$ est un fermé ssi $\forall x \in X$ on a l'équivalence suivante

$$x \in A \iff \forall U \in \Omega X, \quad (x \in U \implies A \cap U \neq \emptyset).$$

Lemme 4.2 (Avec Σ^ω). ▷ Sur Σ^ω , on a que $A \subseteq \Sigma^\omega$ est ouvert ssi $\forall \sigma \in \Sigma^\omega$, on a l'équivalence suivante

$$\sigma \in A \iff \exists \hat{\sigma} \subseteq \sigma, \text{ext}(\hat{\sigma}) \subseteq A.$$

▷ Sur Σ^ω , on a que $A \subseteq \Sigma^\omega$ est fermé ssi $\forall \sigma \in \Sigma^\omega$, on a l'équivalence suivante

$$\sigma \in A \iff \forall \hat{\sigma} \subseteq \sigma, \text{ext}(\hat{\sigma}) \cap A \neq \emptyset,$$

autrement dit,

$$\sigma \in A \iff \forall n \in \mathbb{N}, \left\{ \begin{array}{l} \text{ext}(\sigma(0) \dots \sigma(n)) \cap A \neq \emptyset \\ \Updownarrow \\ \forall \hat{\sigma} \subseteq \sigma, \exists \beta \supseteq \hat{\sigma}, \beta \in A. \end{array} \right.$$

Exemple 4.2. L'ensemble $\{\mathbf{a}\}^\omega$ est un fermé mais pas un ouvert. En effet, si $\hat{\sigma} \subseteq \mathbf{a}^\omega$ alors $\hat{\sigma} = \mathbf{a}^n$, mais, si $|\Sigma| \geq 2$,

$$\text{ext}(\mathbf{a}^n) \not\subseteq \{\mathbf{a}^\omega\}.$$

Corollaire 4.1. Une propriété $P \subseteq (\mathbf{2}^{\text{AP}})^\omega$ est de sûreté ssi P est un fermé.

Preuve. L'idée est que $\text{ext}(P_{\text{bad}})$ est un ouvert et que

$$P = (\mathbf{2}^{\text{AP}})^\omega \setminus \text{ext}(P_{\text{bad}}).$$

□

Proposition 4.1 (Clôture). Soit $A \subseteq X$ où $(X, \Omega X)$ est un espace topologique. Alors,

$$\bar{A} := \bigcap_{A \subseteq C \text{ où } C \text{ fermé}} C$$

est un fermé.

Remarque 4.4. On a que A est un fermé ssi $\bar{A} = A$.

4.1 Théorème de décomposition.

Définition 4.3. Pour $(X, \Omega X)$ un espace topologique, on dit que $A \subseteq X$ est dense si

$$\forall U \in \Omega X, \quad U \neq \emptyset \implies U \cap A \neq \emptyset.$$

Exemple 4.3. Une partie $A \subseteq \Sigma^\omega$ est dense ssi

$$\forall u \in \Sigma^*, \quad \text{ext}(u) \cap A \neq \emptyset,$$

autrement dit, pour tout mot fini $u \in \Sigma^*$, il existe $\sigma \in \Sigma^\omega$ qui étend u (i.e. $u \subseteq \sigma$) et tel que $\sigma \in A$.

Lemme 4.3. On a que $P \subseteq (\mathbf{2}^{\text{AP}})^\omega$ est une propriété de vivacité ssi P est dense.

Théorème 4.1 (Décomposition). Soit $(X, \Omega X)$ un espace et $A \subseteq X$. Alors il existe $C \subseteq X$ un fermé et $D \subseteq X$ dense tel que

$$A = C \cap D.$$

Preuve. On pose $C := \bar{A}$ et $D := A \cup (X \setminus \bar{A})$. Ainsi, on a bien que $A = C \cap D$. On a aussi que C est fermé. Montrons que D est dense.

Soit $U \in \Omega X$ non vide. Si $U \cap A = \emptyset$ alors $A \subseteq X \setminus U$, qui est un fermé. Donc $\bar{A} \subseteq X \setminus U$ et $U \subseteq X \setminus \bar{A}$. □

4.2 Bases.

Définition 4.4. Soit X un ensemble et $\mathcal{B} \subseteq \wp(X)$ tel que \mathcal{B} est stable par intersections finies. Alors,

$$\Omega := \left\{ \bigcup_{i \in I} B_i \mid \forall i \in I, B_i \in \mathcal{B} \right\}$$

est une topologie sur X et \mathcal{B} est appelée *base* de Ω . Autrement dit, on a défini

$$\Omega := \left\{ \bigcup \mathcal{F} \mid \mathcal{F} \subseteq \mathcal{B} \right\}.$$

Lemme 4.4 (Quelques propriétés). ▷ Si $u \subseteq v$ alors $\text{ext}(v) \subseteq \text{ext}(u)$.

▷ Si $|\Sigma| \geq 2$ et $\text{ext}(v) \subseteq \text{ext}(u)$ alors $u \subseteq v$.

(Attention à la contravariance!)

▷ Pour $u, v \in \Sigma^*$, on a

$$\text{ext}(u) \cap \text{ext}(v) = \begin{cases} \text{ext}(v) & \text{si } u \subseteq v \\ \text{ext}(u) & \text{si } v \subseteq u \\ \emptyset & \text{sinon.} \end{cases}$$

Remarque 4.5. Sur Σ^ω , on a que pour tout ouvert U , il existe $W \subseteq \Sigma^*$ tel que $U = \bigcup_{v \in W} \text{ext}(v)$. Avec le lemme précédent, on a que $\Omega \Sigma^\omega$ a pour base

$$\{\text{ext}(u) \mid u \in \Sigma^*\} \cup \{\emptyset\}.$$

Remarque 4.6. On a $\text{ext}(\varepsilon) = \Sigma^\omega$.

Remarque 4.7. L'ensemble Σ^ω est un *espace métrique complet* pour la distance

$$d : \Sigma^\omega \times \Sigma^\omega \longrightarrow [0, 1]$$

$$\alpha, \beta \longmapsto \begin{cases} 0 & \text{si } \alpha = \beta \\ 1/2^{\min n \mid \alpha(n) \neq \beta(n)} & \text{sinon.} \end{cases}$$

On a que $d(\alpha, \gamma) \leq \max(d(\alpha, \beta), d(\beta, \gamma))$.

V. Ordres partiels et treillis.

5.1 Ordres partiels.

Définition 5.1. Un *ordre partiel* (ou *poset* en anglais) est une paire (P, \leq) où \leq est une relation binaire sur P telle que

- ▷ (reflexivité) $\forall x \in P, x \leq x$;
- ▷ (transitivité) $\forall x, y \in P, x \leq y \implies y \leq z \implies x \leq z$;
- ▷ (antisymétrie) $\forall x, y \in P, x \leq y \implies y \leq x \implies x = y$.

Un préordre est une relation binaire reflexive et transitive.

Exemple 5.1. On donne quelques exemples de poset :

1. $(\wp(X), \subseteq)$, l'inclusion dans les parties de X
2. $(\Omega X, \subseteq)$, l'inclusion dans les ouverts de X
3. (Σ^*, \subseteq) , la relation préfixe dans les mots sur Σ

Attention, dans les trois exemples, il existe deux éléments u, v où

$$u \not\leq v \quad \text{et} \quad v \not\leq u.$$

Définition 5.2 (Dual). Soit (P, \leq) un poset. Le *dual* de P est $(P, \leq)^{\text{op}} := (P, \geq)$ où

$$a \geq b \iff b \leq a.$$

Définition 5.3 (Fonction (anti)monotone). Soit (P, \leq_P) et (L, \leq_L) deux posets. Une fonction $f : P \rightarrow L$ est *monotone* si pour tout $a, b \in P$ on a

$$a \leq_P b \implies f(a) \leq_L f(b).$$

On dit que $f : (P, \leq) \rightarrow (L, \leq)$ est *antimonotone* si $f : (P, \geq) = (P, \leq_P)^{\text{op}} \rightarrow (L, \leq_L)$ est monotone, autrement dit pour tout $a, b \in P$ on a

$$a \leq_P b \implies f(a) \geq_L f(b).$$

5.2 Treillis complet.

Définition 5.4. Soit (A, \leq) un poset et $S \subseteq A$.

- ▷ Un *upper bound* de S est un élément $a \in A$ tel que $\forall s \in S, s \leq a$.
- ▷ Un *least upper bound* (*lub*, *join* ou *sup*) de S est un upper bound $a \in A$ de S tel que, pour tout upper bound $b \in A$ de S , on a $a \leq b$.

Par dualité, on a les définitions suivantes.

- ▷ Un élément $a \in A$ est un *lower bound* de S ssi a est un upper bound de S dans A^{op} .

- ▷ Un élément $a \in A$ est un *greatest lower bound* (*glb*, *meet*, *inf*) de S ssi a est un least upper bound de S dans A^{op} .

On note $\bigvee S$ le least upper bound de S . On note $\bigwedge S$ le greatest lower bound de S .

Exemple 5.2. Soit $S \subseteq \wp(X)$ alors le least upper bound de S dans $(\wp(X), \subseteq)$ est $\bigcup S \in \wp(X)$. Le greatest lower bound de S dans $(\wp(X), \subseteq)$ est $\bigcap S \in \wp(X)$.

Exemple 5.3. Soit $S \subseteq \Omega X$ alors le least upper bound dans $(\Omega X, \subseteq)$ est $\bigcup S \in \Omega X$. Le greatest lower bound dans $(\Omega X, \subseteq)$ n'est pas évident. En effet,

$$\{\text{ext}(a^n) \mid n \in \mathbb{N}\} \subseteq \Omega\Sigma^\omega,$$

mais $\bigcap_{n \in \mathbb{N}} \text{ext}(a^n) = \{a^\omega\} \notin \Omega\Sigma^\omega$.

Exemple 5.4. Dans (Σ^*, \subseteq) (la relation « préfixe de »), une partie $S \subseteq \Sigma^*$ n'a pas forcément de sup.

Définition 5.5. Un poset (L, \leq) est un *treillis complet* si

- ▷ tout $S \subseteq L$ a un sup $\bigvee S \in L$;
- ▷ tout $S \subseteq L$ a un inf $\bigwedge S \in L$.

Remarque 5.1 (Unicité du lub/glb). Par antisymétrie, si a et b sont deux least upper bound (ou greatest lower bound) alors $a = b$.

En conséquence on a que tout treillis complet a

- ▷ un plus petit élément $\perp := \bigvee \emptyset \in L$;
- ▷ un plus grand élément $\top := \bigwedge \emptyset \in L$.

Remarque 5.2 (Non-exemple). Le poset (Σ^*, \subseteq) (avec la relation « préfixe de ») n'est *pas* un treillis complet, car il n'a pas de plus grand élément \top .

Exemple 5.5. Le poset $(\wp(X), \subseteq)$ (avec la relation d'inclusion ensembliste) est un treillis complet.

Lemme 5.1. Les conditions suivantes sont équivalentes pour un poset (L, \leq) :

1. (L, \leq) est un treillis complet ;
2. tout $S \subseteq L$ a un sup $\bigvee S \in L$;
3. tout $S \subseteq L$ a un inf $\bigwedge S \in L$;

Preuve. Pour montrer l'implication « 2. \implies 3. », on peut définir

$$\forall S \subseteq L, \quad \bigwedge S := \bigvee \{b \mid \forall s \in S, b \leq s\},$$

et montrer que c'est bien un inf. □

Exemple 5.6. En revenant sur $(\Omega X, \subseteq)$, c'est un treillis complet dont l'inf de $S \subseteq \Omega X$ est

$$\bigwedge S = \bigcup \{V \in \Omega X \mid V \subseteq \bigcap S\}.$$

Il s'agit de $\widehat{\bigcap S}$ qui est l'*intérieur* de $\bigcap S$.

Par exemple, dans $(\Omega\Sigma^\omega, \subseteq)$, on a

$$\bigwedge \{\text{ext}(a^n) \mid n \in \mathbb{N}\} = \widehat{\{a^\omega\}} = \emptyset.$$

- 22/59 -

5.3 Opérateur de clôture.

Définition 5.6. Soit (A, \leq) un poset. Un *opérateur de clôture* sur (A, \leq) est une fonction

$$c : A \rightarrow A$$

telle que

- ▷ c est monotone ;
- ▷ c est « *expansive* » : pour tout $a \in A$, $a \leq c(a)$;
- ▷ c est *idempotent* : $c(c(a)) = c(a)$ pour tout $a \in A$.

Exemple 5.7. Soit $(X, \Omega X)$ un espace topologique. Alors

$$\wp(X) \ni A \mapsto \bar{A} \in \wp(X)$$

est un opérateur de clôture sur $(\wp(X), \subseteq)$.

Lemme 5.2. Soit c un opérateur de clôture sur (L, \leq) . On pose

$$L^c := \{a \in L \mid \underbrace{c(a)}_{a \in \text{im } c} = a\}.$$

Si (L, \leq) est un treillis complet alors (L^c, \leq) est un treillis complet avec

$$\forall S \subseteq L^c, \quad \bigwedge^{L^c} S = \bigwedge^L S.$$

Exemple 5.8. Pour $\overline{(-)} : \wp(X) \rightarrow \wp(X)$ où $(X, \Omega X)$ est un espace topologique, on a

$$(\wp(X))^{\overline{(-)}} = \{F \in \wp(X) \mid F \text{ fermé}\}.$$

Dans ce treillis complet :

$$\bigwedge \mathcal{F} = \bigcap \mathcal{F} \quad \text{et} \quad \bigvee \mathcal{F} = \overline{\bigcup \mathcal{F}},$$

où \mathcal{F} est un ensemble de fermés.

5.4 Connexion de Galois.

Définition 5.7. Considérons deux posets (A, \leq_A) et (B, \leq_B) . Une *connexion de Galois* $g \dashv f : A \rightarrow B$ est une paire (f, g) de fonctions :

$$f : B \rightarrow A \quad \text{et} \quad g : A \rightarrow B$$

telle que

$$g(a) \leq_B b \iff a \leq_A f(b).$$

Exemple 5.9. Soit $f : X \rightarrow Y$ une fonction. On possède deux « lifts » de f sur les powersets :

- ▷ le lift covariant $f_! : \begin{array}{ccc} \wp(X) & \longrightarrow & \wp(Y) \\ A & \longmapsto & \{f(a) \mid a \in A\} \end{array} ;$

- 23/59 -

- ▷ le lift contravariant $f^\bullet : \begin{array}{ccc} \wp(Y) & \longrightarrow & \wp(X) \\ B & \longmapsto & \{x \in X \mid f(x) \in B\} \end{array} .$ ¹

On a que $f_! \dashv f^\bullet$. En effet, pour tout $A \in \wp(X)$ et $B \in \wp(Y)$,

$$\begin{aligned} f_!(A) \subseteq B &\iff \forall x \in X, (x \in A \implies f(x) \in B) \\ &\iff A \subseteq f^\bullet(B). \end{aligned}$$

Exemple 5.10. Soit Σ un alphabet. On a, d'une part,

$$\begin{aligned} \text{Pref} : \wp(\Sigma^\omega) &\longrightarrow \wp(\Sigma^*) \\ A &\longmapsto \underbrace{\{\hat{\sigma} \in \Sigma^* \mid \exists \sigma \in A, \hat{\sigma} \subseteq \sigma\}}_{\bigcup_{\sigma \in A} \text{Pref}(\sigma)}. \end{aligned}$$

D'autre part, on a

$$\begin{aligned} \text{cl} : \wp(\Sigma^*) &\longrightarrow \wp(\Sigma^\omega) \\ W &\longmapsto \{\sigma \in \Sigma^\omega \mid \text{Pref}(\sigma) \subseteq W\}. \end{aligned}$$

Attention, ce n'est pas le cl vu en TD. On a que

$$\text{Pref}(-) \dashv \text{cl}(-).$$

Lemme 5.3. ▷ Si $g \dashv f$ et $g' \dashv f$ alors $g = g'$.

▷ Si $g \dashv f$ et $g \dashv f'$ alors $f = f'$.

▷ Si $g \dashv f$ alors g et f sont monotones.

Preuve. Vu en TD. □

Dans $g \dashv f$, on dit que

- ▷ g est un *adjoint à gauche* de f ;
- ▷ f est un *adjoint à droite* de g .

Lemme 5.4. Si $g \dashv f : (A, \leq_A) \rightarrow (B, \leq_B)$ alors

$$f \circ g : A \xrightarrow{g} B \xrightarrow{f} A$$

est un opérateur de clôture sur (A, \leq_A) .²

Preuve. Vu en TD. □

Exemple 5.11. Pour $\text{Pref}(-) \dashv \text{cl}(-) : \wp(\Sigma^\omega) \rightarrow \wp(\Sigma^*)$, le lemme précédent nous donne l'opérateur de clôture

$$\begin{aligned} \text{cl} \circ \text{Pref} : \wp(\Sigma^\omega) &\longrightarrow \wp(\Sigma^\omega) \\ A &\longmapsto \{\sigma \in \Sigma^\omega \mid \text{Pref}(\sigma) \subseteq \text{Pref}(A)\} \end{aligned}$$

(c'est le $\text{cl}(-)$ vu en TD) est la clôture topologique pour $(\Sigma^\omega, \Omega \Sigma^\omega)$.

¹On note habituellement f^* et non f^\bullet , mais vu qu'on utilise souvent « $*$ » dans le cours, on change de notation.

²Attention à ne pas se tromper sur le sens de la composition !

Remarque 5.3. En particulier, $A \subseteq \Sigma^\omega$ est un fermé si et seulement s'il existe un arbre $T \subseteq \Sigma^*$ tel que

$$A = \{\pi \in \Sigma^\omega \mid \pi \text{ chemin infini dans } T\}.$$

On a que $\text{cl} \circ \text{Pref}(A)$ qui est un arbre sur Σ .

Corollaire 5.1. ▷ Une propriété $P \subseteq (\mathbf{2}^{\text{AP}})^\omega$ est de sûreté si et seulement si on a $P = \text{cl}(\text{Pref}(P))$.

▷ Une propriété $P \subseteq (\mathbf{2}^{\text{AP}})^\omega$ est de vivacité si et seulement si on a $(\mathbf{2}^{\text{AP}})^\omega = \text{cl}(\text{Pref}(P))$.

Preuve. (Déjà) vu en TD. Ceci correspond exactement au fait que

- ▷ P est de sûretéssi P est fermé dans $(\Sigma^\omega, \Omega\Sigma^\omega)$;
- ▷ P est de vivacitéssi P est dense dans $(\Sigma^\omega, \Omega\Sigma^\omega)$;
- ▷ $\text{cl} \circ \text{Pref}$ est exactement $(\overline{-})$ dans $(\Sigma^\omega, \Omega\Sigma^\omega)$.

□

Proposition 5.1. Une propriété $P \subseteq (\mathbf{2}^{\text{AP}})^\omega$ est de vivacité si et seulement si $\text{Pref}(P) = (\mathbf{2}^{\text{AP}})^*$.

Preuve. En effet, par adjonction (connexion de Galois), on a

$$(\mathbf{2}^{\text{AP}})^* = \text{Pref}((\mathbf{2}^{\text{AP}})^\omega) \subseteq \text{Pref}(P) \iff (\mathbf{2}^{\text{AP}})^\omega \subseteq \text{cl}(\text{Pref}(P)).$$

□

Quelques propriétés des connexions de Galois.

Lemme 5.5. Soit $g \dashv f : A \rightarrow B$ une connexion de Galois.

1. pour tout $S \subseteq A$ tel que $\bigvee S \in A$ alors $g(\bigvee S) = \bigvee g_!(S)$;
2. pour tout $S \subseteq B$ tel que $\bigwedge S \in B$ alors $f(\bigwedge S) = \bigwedge f_!(S)$.

Remarque 5.4. Dans le lemme précédent, il est important de remarquer que l'on a une implication « cachée » : $\bigvee S$ existe dans A implique $\bigvee g_!(S)$ existe dans B (et idem pour \bigwedge et $f_!$).

Lemme 5.6. Soient (A, \leq_A) et (B, \leq_B) deux treillis complets.

1. Si $g : A \rightarrow B$ préserve les sups (i.e. $g(\bigvee S) = \bigvee g_!(S)$) alors il existe une fonction $f : B \rightarrow A$ telle que $g \dashv f$. Cette fonction est :

$$f(b) := \bigvee \{a \in A \mid g(a) \leq_B b\}.$$

2. Si $f : B \rightarrow A$ préserve les inf's alors il existe une fonction $g : A \rightarrow B$ telle que $g \dashv f$. Cette fonction est :

$$g(a) := \bigwedge \{b \in B \mid a \leq_A f(b)\}.$$

Exemple 5.12 (Algèbres de Heyting complètes). Soit (L, \leq) un treillis complet. Soit $a \in L$. On a une fonction

$$\begin{aligned} - \wedge a : L &\longrightarrow L \\ b &\longmapsto b \wedge a = \bigwedge \{a, b\}. \end{aligned}$$

On dit que (L, \leq) est une *algèbre de Heyting complète* si, pour tout $a \in A$, la fonction $- \wedge a$ a un adjoint à gauche. Si cet adjoint existe, on le note $a \Rightarrow -$. Ceci nous donne que

$$\forall a, b, c \in L, \quad b \wedge a \leq c \iff b \leq a \Rightarrow c.$$

On a l'équivalence entre :

- ▷ (L, \leq) est une algèbre de Heyting complète ;
- ▷ pour tout $a \in L$, $- \wedge a : L \rightarrow L$ préserve les sups, autrement dit pour tout $S \subseteq L$,

$$(\bigvee S) \wedge a = \bigvee \{s \wedge a \mid s \in S\}.$$

C'est une sorte de distributivité.

Dans ce cas, on a que

$$a \Rightarrow c = \bigvee \{b \mid b \wedge a \leq c\}.$$

VI. Propriétés « observables ».

La terminologie « propriété observable » n'est pas utilisée dans la littérature, mais c'est en réalité la compacité.

Remarque 6.1 (Rappel). Si $f : X \rightarrow Y$ alors

$$f_! \dashv f^* : \wp(X) \rightarrow \wp(Y),$$

où $f_!$ est l'image directe, et f^* est l'image réciproque.

Ainsi, $f^* : \wp(Y) \rightarrow \wp(X)$ préserve les intersections (i.e. si $\mathcal{S} \subseteq \wp(Y)$ alors on a que $f^*(\bigcap \mathcal{S}) = \bigcap_{S \in \mathcal{S}} f^*(S)$).

De plus, f^* préserve les unions car $f^* \dashv f_* : \wp(Y) \rightarrow \wp(X)$ où

$$\begin{aligned} f_* : \wp(X) &\longrightarrow \wp(Y) \\ A &\longmapsto \bigcup \{B \subseteq Y \mid f^*(B) \subseteq A\}. \end{aligned}$$

Définition 6.1. Soient $(X, \Omega X)$ et $(Y, \Omega Y)$ deux espaces topologiques. Une fonction $f : X \rightarrow Y$ est *continue* si $f^* : \wp(Y) \rightarrow \wp(X)$ se restreint en une fonction $f^* : \Omega Y \rightarrow \Omega X$, autrement dit

$$\forall V \in \Omega Y, \quad f^*(V) = \{x \in X \mid f(x) \in V\} \in \Omega(X).$$

On définit ainsi une catégorie d'espaces topologiques.

Un *homéomorphisme* $f : X \rightarrow Y$ est une bijection continue telle que

$$f^{-1} : Y \rightarrow X$$

est continue.¹

Lemme 6.1. Une fonction $f : \Sigma^\omega \rightarrow \Gamma^\omega$ est continue si et seulement si

$$\forall \alpha \in \Sigma^\omega, \forall n \in \mathbb{N}, \exists k \in \mathbb{N}, \forall \beta \in \Sigma^\omega,$$

$$\beta(0) \dots \beta(k) = \alpha(0) \dots \alpha(k)$$

↓

$$f(\beta)(0) \dots f(\beta)(n) = f(\alpha)(0) \dots f(\alpha)(n).$$

Autrement dit, f est continue si et seulement si on peut déterminer une partie finie de sa sortie à partir d'une partie finie de son entrée.

Soit $P \subseteq \Sigma^\omega$, et on définit la *fonction caractéristique* de P :

$$\chi_P : \Sigma^\omega \rightarrow \mathbf{2} = \{0, 1\}$$

$$\alpha \mapsto \begin{cases} 1 & \text{si } \alpha \in P \\ 0 & \text{si } \alpha \notin P \end{cases}.$$

Avec $\Omega \mathbf{2} = \wp(\mathbf{2}) = \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$ (ce qui est cohérent avec l'idée que $\mathbf{2}$ représente les booléens), on a que χ_P est continue si et seulement si

- ▷ $\chi_P^{-1}\{0\} = \Sigma^\omega \setminus P$ est un ouvert ;
- ▷ $\chi_P^{-1}\{1\} = P$ est un ouvert.

On arrive donc à la notion de *clopen*.

Définition 6.2. Soit $(X, \Omega X)$ un espace topologique. Une partie $P \subseteq X$ est *clopen* (*ouvert fermé* en français) si P et $X \setminus P$ sont ouverts.

Remarque 6.2. 1. On a \emptyset est clopen, et que, si A et B sont clopen alors $A \cup B$ est clopen.

2. On a X est clopen, et que, si A et B sont clopen alors $A \cap B$ est clopen (dual du point précédent).

3. Si A est clopen alors $X \setminus A$ est clopen.

Exemple 6.1. Soit $u \in \Sigma^*$, on a que $\text{ext}(u)$ est ouvert. Mais, on a aussi que $\Sigma^\omega \setminus \text{ext}(u)$ est ouvert :

$$\Sigma^\omega \setminus \text{ext}(u) = \bigcup \{\text{ext}(v) \mid v \neq u \text{ et } \text{length}(v) = \text{length}(u)\}.$$

¹Ce n'est pas évident : par exemple, il y a une bijection $[0, 1] \rightarrow S^1$ (où S^1 est le cercle unité de \mathbb{R}^2) continue mais la réciproque ne l'est pas.

Remarque 6.3. Sur $(\Sigma^\omega, \Omega \Sigma^\omega)$, tous les $\text{ext}(W)$ où $W \subseteq \Sigma^*$ est *fini* sont clopen. La réciproque est fausse, comme le montre le lemme suivant.

Lemme 6.2. Si Σ est infini et $a \in \Sigma$, alors

$$\Sigma^\omega \setminus \text{ext}(a) = \bigcup_{\Sigma \ni b \neq a} \text{ext}(b)$$

est clopen mais *pas* de la forme $\text{ext}(W)$ avec W fini.

6.1 Compacité.

Définition 6.3. Soit $(X, \Omega X)$ un espace topologique.

1. Une partie $A \subseteq X$ est *compacte* si, pour toute famille $(V_i)_{i \in I} \in \Omega X^I$ telle que $A \subseteq \bigcup_{i \in I} V_i$, il existe $J \subseteq I$ *fini* tel que $A \subseteq \bigcup_{j \in J} V_j$.
2. On dit que $(X, \Omega X)$ est *compact* si X est une partie compacte.

Remarque 6.4 (Non-exemple). Si Σ est infini alors Σ^ω n'est pas compact :

$$\Sigma^\omega = \bigcup_{a \in \Sigma} \text{ext}(a).$$

Proposition 6.1. Si Σ est *fini* alors Σ^ω est compact.

Preuve. On procède à l'aide du lemme de König. Supposons que $\Sigma^\omega = \bigcup_{i \in I} U_i$ où $U_i \in \Omega \Sigma^\omega$. On a que $U_i = \text{ext}(V_i)$ pour un $V_i \subseteq \Sigma^*$ (en général, V_i est infini). Soit $V = \bigcup_{i \in I} V_i \subseteq \Sigma^*$, et on vérifie que $\text{ext}(V) = \Sigma^\omega$. Pour chaque $n \in \mathbb{N}$, on définit $W_n \subseteq \Sigma^n$ par récurrence :

▷ On pose $W_0 := \{\varepsilon\}$ si $\varepsilon \in V$ et $W_0 := \emptyset$ sinon.

▷ On pose

$$W_{n+1} := \{u \in V \mid u \text{ n'a pas de préfixe dans } \bigcup_{k \leq n} W_k\}.$$

On pose enfin $W = \bigcup_{n \in \mathbb{N}} W_n$. On a que $\text{ext}(W) = \text{ext}(V)$ (car, pour tout $v \in V$, il existe $w \in W$ tel que $w \subseteq v$), et W est « prefix-free » (c'est-à-dire que, pour $w, w' \in W$, on a $w \not\subseteq w'$ ssi $w \neq w'$).

Si W est fini alors on s'arrête.

Par l'absurde, supposons W infini, et posons $T = \text{Pref}(W)$ qui est un arbre par définition. L'arbre T est à branchement fini (car Σ est fini), et T est infini (car W l'est). Par le lemme de König, il existe un chemin infini $\pi \in \Sigma^\omega$ dans T . Comme $\Sigma^\omega = \text{ext}(W)$, il existe $u \in W$ tel que $u \subseteq \pi$. De plus, il existe $a \in \Sigma$ tel que $ua \subseteq \pi$ et donc $ua \in T = \text{Pref}(W)$.

On arrive à une contradiction car $u \in W$ et W est prefix-free. □

Corollaire 6.1. On a que Σ^ω est compact si et seulement si Σ est fini.

Lemme 6.3. Si $(X, \Omega X)$ est compact et $C \subseteq X$ est fermé alors C est compact.

Preuve. L'idée est que si $C \subseteq \bigcup_{i \in I} V_i$ alors $X \subseteq (X \setminus C) \cup \bigcup_{i \in I} V_i$. □

Corollaire 6.2. Si Σ est fini alors $A \subseteq \Sigma^\omega$ est clopen si et seulement si $A = \text{ext}(W)$. □

6.2 Espace Hausdorff.

Définition 6.4. On dit que $(X, \Omega X)$ est *Hausdorff* (ou T_2) lorsque, pour tout $x \neq y \in X$, alors il existe $U, V \in \Omega X$ tels que

$$U \cap V = \emptyset \quad x \in U \quad \text{et} \quad y \in V.$$

Exemple 6.2. L'espace $(\Sigma^\omega, \Omega\Sigma^\omega)$ est Hausdorff. Soient $\alpha \neq \beta$. Il existe $u \subseteq \alpha$ et $v \subseteq \beta$ tels que $\text{ext}(u) \cap \text{ext}(v) = \emptyset$.

(On peut choisir $u = p\alpha(\text{length}(p))$ et $v = p\beta(\text{length}(p))$ où p est le plus long préfixe commun à α et β .)

Proposition 6.2. Si (X, Ω) est compact Hausdorff et $C \subseteq X$ est compact alors C est fermé.

Preuve. Soit $C \subseteq X$ est compact et $x \notin C$. Pour tout $y \in C$, il existe U_y, V_y tels que $U_y \cap V_y = \emptyset$ et $x \in U_y$ et $y \in V_y$. Donc, on a $C \subseteq \bigcup_{y \in C} V_y$. Comme C est compact, il existe $y_1, \dots, y_n \in C$ tels que $C \subseteq V_{y_1} \cup V_{y_2} \cup \dots \cup V_{y_n}$. On a que $x \in U_{y_1} \cup \dots \cup U_{y_n} =: U \in \Omega X$. Et, $U \subseteq X \setminus C$ car, $U \cap (V_{y_1} \cup \dots \cup V_{y_n}) = \emptyset$. \square

Corollaire 6.3. Si $(X, \Omega X)$ est compact Hausdorff et $C \subseteq X$,

$$C \text{ compact} \iff C \text{ fermé.}$$

VII. Logique temporelle linéaire.

7.1 La logique LML.

Remarque 7.1 (Idée). La signification de LML est *linear-time modal logic*. L'idée est de définir une logique pour une propriété LT $P \subseteq (\mathbf{2}^{\text{AP}})^\omega$ telle que, pour AP fini, les formules correspondent aux clopens.

Soit AP un ensemble de proposition atomiques.

Définition 7.1. Les formules de LML sont

$$\begin{array}{ll} \phi, \psi ::= a & a \in \text{AP} \\ | \text{True} & (\text{parfois notée } \top) \\ | \text{False} & (\text{parfois notée } \perp) \\ | \phi \wedge \psi \\ | \phi \vee \psi \\ | \neg \phi \\ | \circ \phi \\ . \end{array}$$

La modalité \circ est appelée *later* ou *next*.

Définition 7.2. L'interprétation $[\![\phi]\!] \subseteq (\mathbf{2}^{\text{AP}})^\omega$ est définie par :

- ▷ $[\![a]\!] := \{\sigma \mid a \in \sigma(0)\};$
- ▷ $[\![\text{True}]\!] := (\mathbf{2}^{\text{AP}})^\omega;$
- ▷ $[\![\phi \wedge \psi]\!] := [\![\varphi]\!] \cap [\![\psi]\!];$
- ▷ $[\![\text{False}]\!] := \emptyset;$
- ▷ $[\![\phi \vee \psi]\!] := [\![\varphi]\!] \cup [\![\psi]\!];$
- ▷ $[\![\neg \phi]\!] := (\mathbf{2}^{\text{AP}})^\omega \setminus [\![\phi]\!];$
- ▷ $[\![\circ \phi]\!] := \{\sigma \mid \sigma \upharpoonright 1 \in [\![\phi]\!]\}$ où, pour $\sigma \in (\mathbf{2}^{\text{AP}})^\omega$, on note

$$\begin{aligned} \sigma \upharpoonright i : \mathbb{N} &\longrightarrow \mathbf{2}^{\text{AP}} \\ k &\longmapsto \sigma \upharpoonright \sigma(k+i), \end{aligned}$$

(c'est un décalage d'indices).

Exemple 7.1. Quelques exemples de mots tels que $\sigma \in [\![a \vee \circ b]\!]$:

- ▷ $\sigma = \{a\}\emptyset^\omega,$
- ▷ $\sigma = \{b\}\{a, b\}^\omega,$
- ▷ $\sigma = \emptyset\{a, b\}\emptyset^\omega.$

Proposition 7.1. Pour ϕ une formule de LML, on a que $[\![\phi]\!]$ est clopen dans $(\mathbf{2}^{\text{AP}})^\omega$.

Preuve. Par induction sur ϕ , on a les cas suivants.

- ▷ On a que $[\![a]\!] = \bigcup_{a \in A \subseteq \text{AP}} \text{ext}(A)$ ¹ est un ouvert. De plus, on a que $(\mathbf{2}^{\text{AP}})^\omega \setminus [\![a]\!] = \bigcup_{a \notin B \subseteq \text{AP}} \text{ext}(B)$ est un ouvert.
- ▷ On a que
 - $[\![\circ \phi]\!] = \bigcup_{u \in W, A \subseteq \text{AP}} \text{ext}(Au)$
 - $(\mathbf{2}^{\text{AP}})^\omega \setminus [\![\circ \phi]\!] = \bigcup_{v \in V, A \subseteq \text{AP}} \text{ext}(Av)$

où par hypothèse d'induction, il existe $V, W \subseteq \Sigma^*$ tels que $[\![\phi]\!] = \text{ext}(W)$ et $(\mathbf{2}^{\text{AP}})^\omega \setminus [\![\phi]\!] = \text{ext}(V)$.

- ▷ De même pour les autres cas.

□

Proposition 7.2. Si AP est *fini* et $P \subseteq (\mathbf{2}^{\text{AP}})^\omega$ est clopen alors il existe ϕ une formule LML telle que $P = [\![\phi]\!]$.

Preuve. On a que $P = \text{ext}(W)$ où $W \subseteq (\mathbf{2}^{\text{AP}})^*$ est *fini*. On a montrer par récurrence sur la taille du mot u que :

$$\forall u \in (\mathbf{2}^{\text{AP}})^*, \quad \exists \phi_u \text{ une formule LML}, \quad [\![\phi_u]\!] = \text{ext}(u).$$

¹On rappelle que $\text{ext}(A) = \{\sigma \mid \sigma(0) = A \subseteq \text{AP}\}$.

▷ Cas de base. Soit $A \subseteq AP$, on peut prendre

$$\phi_A := (\bigwedge_{a \in A} a) \wedge (\bigwedge_{b \notin A} \neg b).$$

▷ Récurrence. Soit $u = Av$ où $A \subseteq AP$ et $v \in (2^{AP})^*$. On peut poser

$$\phi_{Av} := \phi_A \wedge \circ \phi_v.$$

On peut aussi faire le cas de base pour ε , en posant $\phi_\varepsilon := \text{True}$. \square

Corollaire 7.1. Si AP est fini et $P \subseteq (2^{AP})^\omega$ alors

$$P \text{ clopen} \iff \text{il existe } \phi \text{ telle que } \llbracket \phi \rrbracket = P.$$

\square

7.1.1 Équivalences logiques.

Définition 7.3. On note $\phi \equiv \psi$ lorsque $\llbracket \phi \rrbracket = \llbracket \psi \rrbracket$. On dit que ϕ et ψ sont (*logiquement*) équivalentes.

On a les équivalences suivantes :

- ▷ $\phi \equiv \phi \wedge \phi$
- ▷ $\phi \equiv \text{True} \wedge \phi$
- ▷ $\text{True} \equiv \phi \vee \neg \phi$
- ▷ $\text{False} \equiv \phi \wedge \neg \phi$
- ▷ $\phi \equiv \neg \neg \phi$
- ▷ $\circ(\phi \vee \psi) \equiv \circ \phi \vee \circ \psi$
- ▷ $\circ \text{False} \equiv \text{False}$
- ▷ $\circ(\phi \wedge \psi) \equiv \circ \phi \wedge \circ \psi$
- ▷ $\circ \text{True} \equiv \text{True}$

d'autres équivalences sont possibles (c.f. figure 6 des notes de cours).

7.1.2 Homework : Dualité de Stone.

L'idée est de motiver le DM, et de donner quelques bases sur ce que l'on va montrer. On se place dans le cas où AP est un ensemble fini. Considérons un mot $\sigma \in (2^{AP})^\omega$, on pose

$$\mathcal{F}_\sigma := \{[\phi]_\equiv \mid \sigma \in \llbracket \phi \rrbracket\},$$

où $[\phi]_\equiv$ est la classe d'équivalence de \equiv . Par les résultats précédents, on a que

$$\mathcal{F}_\sigma \cong \{C \mid \sigma \in C \text{ clopen}\}.$$

On a $\sigma \neq \beta$ implique $\mathcal{F}_\sigma \neq \mathcal{F}_\beta$. De plus, on a les propriétés suivantes :

1. si $C \in \mathcal{F}_\sigma$ et $C \subseteq D$ clopen alors $D \in \mathcal{F}_\sigma$;

2. si $C, D \in \mathcal{F}_\sigma$ alors $C \cap D \in \mathcal{F}_\sigma$;

3. $(2^{AP})^\omega \in \mathcal{F}_\sigma$;

4. si C, D sont clopen tels que $C \cup D \in \mathcal{F}_\sigma$ alors $C \in \mathcal{F}_\sigma$ ou $D \in \mathcal{F}_\sigma$;

5. $\emptyset \in \mathcal{F}_\sigma$.

Ces cinq propriétés caractérisent totalement les mots infinis, comme le montre le théorème suivant.

Théorème 7.1. Si \mathcal{F} est un ensemble de clopens dans $(2^{AP})^\omega$ vérifiant les cinq propriétés, alors il existe $\sigma \in (2^{AP})^\omega$ tel que $\mathcal{F} = \mathcal{F}_\sigma$.

Ce théorème est une spécialisation de la *dualité de Stone*.

Définition 7.4. On dit que $(X, \Omega X)$ est un *espace de Stone* s'il est compact, Hausdorff et qu'il admet une base de clopens.

Théorème 7.2. Si $(X, \Omega X)$ est un espace de Stone alors

$$(X, \Omega X) \cong \mathbf{Sp}(\underbrace{\{C \mid C \text{ clopen}}_{\text{algèbre de Boole}}, \subseteq).$$

On va définir le *spectre* $\mathbf{Sp}(B)$ où B est une algèbre de Boole, à l'aide des ultrafiltres sur B et les filtres premiers (c.f. les cinq propriétés ci-dessus).

Remarque 7.2 (Idée). Si $\mathcal{F} \subseteq B$ alors c'est une « théorie consistante et complète » où

- ▷ *théorie* : stable par implication, c.f. 1.-3.
- ▷ *consistante* : sans contradiction, 5.
- ▷ *complète* : 4., $\forall C$ clopen, si $C \notin \mathcal{F}_\sigma$ alors $(2^{AP})^\omega \setminus C \in \mathcal{F}_\sigma$.

7.1.3 Extension de LML avec \square et \diamond .

Remarque 7.3. Avec LML, on ne définit que des clopens. Ainsi, les propriétés de sécurité ne sont que « finitaires » et il n'y a que $(2^{AP})^\omega$ comme propriété de vivacité.

Définition 7.5. On ajoute à LML les modalités

- ▷ $\square \phi$ qui signifie « always » (notée parfois $A\phi$) ;
- ▷ $\diamond \phi$ qui signifie « eventually » (notée parfois $E\phi$) ;

où

- ▷ $\llbracket \square \phi \rrbracket := \{ \sigma \in (2^{AP})^\omega \mid \forall n \in \mathbb{N}, \sigma \upharpoonright n \in \llbracket \phi \rrbracket \}$;
- ▷ $\llbracket \diamond \phi \rrbracket := \{ \sigma \mid \exists n \in \mathbb{N}, \sigma \upharpoonright n \in \llbracket \phi \rrbracket \}$.

Dans la suite, les preuves se termineront par « QED » au lieu du symbole usuel « \square », pour enlever l'ambiguité avec la modalité.

Notation. On note $\sigma \Vdash \phi$ pour $\sigma \in \llbracket \phi \rrbracket$.

Exemple 7.2. On a

1. $\sigma \Vdash \Diamond a$ ssi $\exists n \in \mathbb{N}, a \in \sigma(n)$, l'ensemble $\llbracket \Diamond a \rrbracket$ est ouvert mais pas compact, donc pas un clopen car

$$\llbracket \Diamond a \rrbracket = \bigcup_{u \in (\mathbf{2}^{\text{AP}})^*, a \in A} \text{ext}(uA);$$

2. $\sigma \Vdash \Box a$ ssi $\forall n \in \mathbb{N}, a \in \sigma(n)$ (l'ensemble $\llbracket \Box a \rrbracket$ est un fermé non clopen);
 3. $\sigma \Vdash \Box \Diamond a$ ssi $\exists^{\infty} n, a \in \sigma(n)$ (c'est une propriété de vivacité);
 4. $\sigma \Vdash \Diamond \Box a$ ssi $\forall^{\infty} n, a \in \sigma(n)$ (c'est une propriété de vivacité).

Lemme 7.1. On a que :

1. $\Box \phi \equiv \neg \Diamond \neg \phi$;
2. $\Diamond \phi \equiv \neg \Box \neg \phi$;
3. $\Box \phi \equiv \phi \wedge \Box \phi$;
4. $\Diamond \phi \equiv \phi \vee \Box \phi$.

Avec des \wedge et des \vee infinis, on aurait :

$$\Box \phi \equiv \phi \wedge \Box \phi \wedge \Box \Box \phi \wedge \dots \equiv \bigwedge_{n \in \mathbb{N}} \Box^n \phi$$

et

$$\Diamond \phi \equiv \phi \vee \Diamond \phi \vee \Diamond \Diamond \phi \vee \dots \equiv \bigvee_{n \in \mathbb{N}} \Diamond^n \phi.$$

Définition 7.6. On étend ensuite LML par variables :

$$\phi, \psi ::= X \mid \dots,$$

où $X, Y, \dots \in \mathcal{X}$. Une *valuation* de \mathcal{X} est une fonction

$$\rho : \mathcal{X} \rightarrow \wp((\mathbf{2}^{\text{AP}})^\omega),$$

où la sémantique $\llbracket \phi \rrbracket_\rho$ est définie de manière très similaire à $\llbracket \phi \rrbracket$ en ajoutant

$$\llbracket X \rrbracket_\rho := \rho(X).$$

Notation. Soient ϕ, ρ, X , on pose

$$\begin{aligned} \llbracket \phi \rrbracket_\rho(X) &: \wp((\mathbf{2}^{\text{AP}})^\omega) \longrightarrow \wp((\mathbf{2}^{\text{AP}})^\omega) \\ A &\longmapsto \llbracket \phi \rrbracket_\rho[A/X], \end{aligned}$$

où $\rho[A/X](X) = A$ et $\rho[A/X](Y) = \rho(Y)$ si $Y \neq X$.

Lemme 7.2. Soit ϕ et X n'apparaissant pas dans ϕ . On pose

$$\phi_\Diamond(X) := \phi \wedge \Diamond X \quad \phi_\Box(X) := \phi \wedge \Box X.$$

Alors, pour tout ρ ,

- ▷ $\llbracket \Diamond \phi \rrbracket_\rho$ est le plus petit point fixe de $\llbracket \phi_\Diamond \rrbracket_\rho(X)$;
 - 33/59 -

▷ $\llbracket \Box \phi \rrbracket_\rho$ est le plus grand point fixe de $\llbracket \phi_\Box \rrbracket_\rho(X)$.

Preuve. Pour le premier point, on doit montrer les deux propriétés suivantes :

1. $\llbracket \Diamond \phi \rrbracket_\rho = \llbracket \phi_\Diamond \rrbracket(\llbracket \Diamond \phi \rrbracket)$;
2. $\forall P \subseteq (\mathbf{2}^{\text{AP}})^\omega, P = \llbracket \phi_\Diamond \rrbracket_\rho(P) \implies \llbracket \Diamond \phi \rrbracket_\rho \subseteq P$.

Pour la première propriété, c'est la loi d'expansion de \Diamond . Pour la seconde, soit P tel que $P = \llbracket \phi \vee \Diamond X \rrbracket(P)$ et $\sigma \in \llbracket \Diamond \phi \rrbracket$, et montrons que $\sigma \in P$. Soit $n \in \mathbb{N}$ tel que $\sigma \restriction n \in \llbracket \phi \rrbracket$. Alors, on a que $\sigma \restriction n \in \llbracket \phi \vee \Diamond X \rrbracket(P)$ et donc $\sigma \restriction n \in P$. Donc $\sigma \restriction (n-1) \in \llbracket \phi \vee \Diamond \phi \rrbracket(P) = P$, donc pour tout $i \leq n$, $\sigma \restriction i \in \llbracket \phi \vee \Diamond X \rrbracket(P) = P$ et en particulier $\sigma = \sigma \restriction 0 \in P$. **QED**

Définition 7.7. On dit qu'une variable X est *positive* dans ϕ si toutes les occurrences de X dans ϕ sont sous un nombre pair de \neg . (On peut donner une définition inductive de « X est positive dans ϕ » et de « ϕ est négative dans ϕ ».)

Exemple 7.3. Dans les exemples ci-dessous, on a que X est positive dans ϕ :

$$a \wedge \Diamond X \quad a \vee \Diamond X \quad \text{et } \neg(\neg a \wedge \Diamond \neg X),$$

mais, dans les exemples ci-dessous, X n'est pas positive dans ϕ :

$$\neg X \quad \text{et} \quad \neg X \vee (a \wedge \Diamond X).$$

Lemme 7.3. Si X est positive dans ϕ alors

$$\llbracket \phi \rrbracket(X) : (\wp((\mathbf{2}^{\text{AP}})^\omega), \subseteq) \longrightarrow (\wp((\mathbf{2}^{\text{AP}})^\omega), \subseteq)$$

est monotone. **QED**

7.1.4 Théorème de Knaster-Tarski.

Définition 7.8. Soient (L, \leq) un poset et $f : L \rightarrow L$.

1. On dit que $a \in L$ est un *pré-pointfixe* de f si $f(a) \leq a$.
2. On dit que $a \in L$ est un *post-pointfixe* de f si $a \leq f(a)$.

Théorème 7.3 (Knaster-Tarski). Soit $f : L \rightarrow L$ une fonction monotone où (L, \leq) est un treillis complet. Alors

$$\mu(f) := \bigwedge \{a \in L \mid f(a) \leq a\}$$

est le plus petit point fixe de f , et

$$\nu(f) := \bigvee \{a \in L \mid a \leq f(a)\}$$

est le plus grand point fixe de f . **QED**

Preuve. Vu en TD.

Lemme 7.4. Soit $f : (\wp(X), \subseteq) \rightarrow (\wp(X), \subseteq)$ monotone. On pose

$$\begin{aligned} g : \wp(X) &\longrightarrow \wp(X) \\ A &\longmapsto g(A) = X \setminus f(X \setminus A). \end{aligned}$$

On a alors que

$$\begin{aligned} \mu(f) &= X \setminus \nu(g) & \nu(f) &= X \setminus \mu(g). \\ - 34/59 - \end{aligned}$$

Preuve.

$$\begin{aligned}
X \setminus \nu(g) &= X \setminus \bigcup \{A \mid A \subseteq g(a)\} \\
&= \bigcap \{X \setminus A \mid A \subseteq g(A)\} \\
&= \bigcap \{X \setminus A \mid A \subseteq X \setminus f(X \setminus A)\} \\
&= \bigcap \{X \setminus A \mid f(X \setminus A) \subseteq X \setminus A\} \\
&= \bigcap \{B \mid f(B) \subseteq B\} \\
&= \nu(f).
\end{aligned}$$

QED

7.2 La logique LTL.

La logique LTL est une extension de LML par certains points fixes. Soit $\theta(X)$ avec X positive dans θ . On peut écrire

$$\theta(X) \equiv \psi \vee \left(\bigwedge_{i \in I} \left(\phi_i \wedge \bigwedge_{j \in J_i} \circ^{n_{i,j}} X \right) \right),$$

où X n'apparaît pas dans ψ ni dans ϕ_i pour tout $i \in I$.

Dans LTL, on va avoir les points fixes de θ si $n_{i,j} = 1$. Dans ce cas, on pourra écrire

$$\theta(X) \equiv \psi \vee (\phi \wedge \circ X),$$

où X n'apparaît pas dans ψ ni dans ϕ .

Définition 7.9. On définit la syntaxe de LTL par

$$\begin{array}{ll}
\phi, \psi ::= a & a \in AP \\
| \phi \wedge \psi \\
| \phi \vee \psi \\
| \text{False} \\
| \text{True} \\
| \neg \phi \\
| \circ \phi \\
| \phi \text{ Until } \psi \\
\cdot
\end{array}$$

La modalité ϕ Until ψ est notée $\phi \mathbf{U} \psi$ dans les notes de cours.

La sémantique pour LTL est définie de manière identique à la sémantique de LML en ajoutant

$$\llbracket \phi \text{ Until } \psi \rrbracket = \left\{ \sigma \in (2^{\text{AP}})^\omega \mid \exists i \in \mathbb{N}, \quad \begin{array}{l} \sigma \restriction i \in \llbracket \psi \rrbracket \\ \forall j < i, \sigma \restriction j \in \llbracket \phi \rrbracket \end{array} \right\}.$$

7.2.1 Points fixes dans LTL.

On étend, dans cette sous-section uniquement, LTL avec des variables (comme pour LML).

Lemme 7.5. Soit X n'apparaissant pas dans ϕ ni ψ . Alors, on a que $\llbracket \phi \text{ Until } \psi \rrbracket$ est le plus petit point fixe de $\llbracket \theta \rrbracket(X)$.

Preuve. On a

$$\phi \text{ Until } \psi \equiv \psi \vee (\phi \wedge \circ(\phi \text{ Until } \psi)).$$

Soit P tel que $P = \llbracket \theta \rrbracket(P)$. Soit $\sigma \in \llbracket \phi \text{ Until } \psi \rrbracket$. Soit $i \in \mathbb{N}$ tel que $\sigma \restriction i \in \llbracket \psi \rrbracket$ et $\sigma \restriction 0, \dots, \sigma \restriction (i-1) \in \llbracket \phi \rrbracket$. On a $\sigma \restriction i \in \llbracket \theta \rrbracket(P) = P$, et comme $\sigma \restriction (i-1) \in \llbracket \phi \rrbracket$, on a que $\sigma \restriction (i-1) \in \llbracket \theta \rrbracket(P) = P$. Ainsi, pour tout $k \in \{i-1, \dots, 0\}$, on a $\sigma \restriction k \in \llbracket \theta \rrbracket(P) = P$, donc $\sigma = \sigma \restriction 0 \in P$. **QED**

Remarque 7.4. Si X est (positive et) sous exactement un \circ dans $\theta(X)$ alors c'est le cas aussi dans $\neg\theta[\neg X/X]$.

On a

$$\begin{aligned}
\neg\theta[\neg X/X] &= \neg(\phi \vee (\psi \wedge \circ(\neg X))) \\
&\equiv \neg\psi \wedge (\neg\phi \vee \circ X) \\
&\equiv (\neg\psi \wedge \neg\phi) \vee (\neg\psi \wedge \circ X)
\end{aligned}$$

donc

$$\mu \llbracket \neg\theta[\neg X/X] \rrbracket(X) = \llbracket \neg\psi \text{ Until } (\neg\psi \wedge \neg\phi) \rrbracket.$$

Lemme 7.6. Soit X n'apparaissant pas dans ϕ, ψ . Alors

$$\llbracket \neg(\neg\psi \text{ Until } (\neg\psi \wedge \neg\phi)) \rrbracket$$

qui est un plus petit point fixe de $\llbracket \theta \rrbracket(X)$ pour $\theta(X) = \psi \vee (\phi \wedge \circ X)$.

Notation. On note

$$\phi \text{ WUntil } \psi := \neg(\neg\psi \text{ Until } (\neg\psi \wedge \neg\phi)),$$

pour *weak until*.

Notation. On note

$$\diamond \phi := \text{True Until } \phi \quad \square \phi := \phi \text{ WUntil False},$$

et ont la même sémantique que pour LML.

Remarque 7.5. On peut montrer que le plus grand point fixe de $\theta(X) = a \wedge \circ \circ X$ n'est pas définissable dans LTL.

7.2.2 Équivalences logiques pour LTL.

On note, comme avant, $\phi \equiv \psi$ si $\llbracket \phi \rrbracket = \llbracket \psi \rrbracket$. On a des lois, comme pour LML, *c.f.* figure 8 dans les notes de cours (section 7.3.3).

Exemple 7.4. On a

$$\begin{aligned}
\diamond \text{False} &\equiv \text{False} \\
\diamond(\phi \vee \psi) &\equiv \diamond \phi \vee \diamond \psi \\
\square \text{True} &\equiv \text{True} \\
\square(\phi \wedge \psi) &\equiv \square \phi \wedge \square \psi \\
\square(\phi \text{ Until } \psi) &\equiv \square \phi \text{ Until } \square \psi
\end{aligned}$$

Lemme 7.7. On a

1. $\neg(\phi \text{ WUntil } \psi) \equiv \neg\psi \text{ Until } (\neg\psi \wedge \neg\phi)$;
2. $\neg(\phi \text{ Until } \psi) \equiv \neg\psi \text{ WUntil } (\neg\psi \wedge \neg\phi)$.

QED

Lemme 7.8. On a

$$\phi \text{ WUntil } \psi \equiv (\phi \text{ Until } \psi) \vee \square \phi.$$

Preuve. Soit $P = \llbracket(\phi \text{ Until } \psi) \vee \square \phi\rrbracket$. On va montrer que $P = \nu(\llbracket\theta\rrbracket(X))$ où $\theta = \psi \vee (\phi \wedge \square X)$. On a que $P = \llbracket\theta\rrbracket(P)$ (c'est « facile »). Soit Q tel que $Q = \llbracket\theta\rrbracket(Q)$ et on va montrer que $Q \subseteq P$. Soit $\sigma \in Q$.

▷ Si $\sigma \in \llbracket\square\phi\rrbracket$ alors $\sigma \in P$ par définition de P .

▷ Sinon soit i le plus petit tel que $\sigma \upharpoonright i \notin \llbracket\phi\rrbracket$. Ainsi, par définition, $\sigma \upharpoonright j \in \phi$ pour $j < i$. On montre qu'il existe un certain $k \leq i$ tel que $\sigma \upharpoonright k \in \llbracket\psi\rrbracket$. Par l'absurde, supposons que $\sigma \upharpoonright k \notin \llbracket\psi\rrbracket$ pour tout $k \leq i$. On voit que pour tout $k < i$, on a $\sigma \upharpoonright k \in \llbracket\theta\rrbracket(Q) = Q$ donc $\sigma \upharpoonright (i-1) \in \llbracket\theta\rrbracket(Q) = Q$, et donc $\sigma \upharpoonright (i-1) \notin \llbracket\psi\rrbracket$, ce qui implique $\sigma \upharpoonright i \in Q$, ce qui contredit la définition de i (car $\sigma \upharpoonright i \notin \llbracket\psi\rrbracket$ et $\sigma \upharpoonright i \notin \llbracket\phi\rrbracket$).

QED

VIII. Langages ω -réguliers.

Cette partie du cours n'est pas dans les notes de cours mais la source est

Baier, C. and Katoen, J.-P., *Principles of Model Checking*, MIT Press, 2008.

8.1 Expressions ω -régulières.

L'objet de cette section est l'analogue des expressions régulières pour les ω -mots. Soit Σ un alphabet non vide et fini.

Définition 8.1 (Quelques opérations). Soit F une expression régulière sur Σ telle que $\varepsilon \notin \mathcal{L}(F)$. On pose

$$\mathcal{L}(F)^\omega := \left\{ \sigma \in \Sigma^\omega \mid \begin{array}{l} \sigma = v_1 \cdot v_2 \cdot v_3 \cdot \dots \\ \forall i \geq 1, v_i \in \mathcal{L}(F) \end{array} \right\}.$$

Soit, de plus, E une expression régulière sur Σ . On pose

$$\mathcal{L}(E) \cdot (\mathcal{L}(F))^\omega := \left\{ u \cdot \sigma \in \Sigma^\omega \mid \begin{array}{l} u \in \mathcal{L}(E) \\ \sigma \in \mathcal{L}(F)^\omega \end{array} \right\}.$$

Attention. On peut avoir $u = \varepsilon \in \mathcal{L}(E)$.

– 37/59 –

Définition 8.2. Une *expression ω -régulière* sur Σ est de la forme

$$G := E_1 \cdot F_1^\omega + \dots + E_n F_n^\omega,$$

où les E_i, F_i soin des expressions régulières (classiques) sur Σ ,¹ avec $\varepsilon \notin \mathcal{L}(F_i) \subseteq \Sigma^*$. Le langage de G est

$$\mathcal{L}_\omega(G) := \mathcal{L}(E_1) \cdot (\mathcal{L}(F_1))^\omega \cup \dots \cup \mathcal{L}(E_n) \cdot (\mathcal{L}(F_n))^\omega \subseteq \Sigma^\omega.$$

Exemple 8.1. Avec $\Sigma = \{a, b\}$, on a

$$P := (b^* a)^\omega \equiv \{\varepsilon\} \cdot (b^* a)^\omega \equiv (b^* a) \cdot (b^* a)^\omega,$$

où l'on note $G \equiv G'$ quand $\mathcal{L}_\omega(G) = \mathcal{L}_\omega(G')$.

On a que

$$\mathcal{L}_\omega(P) = \{\sigma \in \Sigma^\omega \mid \exists^\infty t, \sigma(t) = a\}.$$

Exemple 8.2. Avec $\Sigma = \{a, b\}$ et $S := (a+b)^* \cdot b^\omega$, on a

$$\mathcal{L}_\omega(S) = \{\sigma \in \Sigma^\omega \mid \forall^\infty t, \sigma(t) = b\}.$$

8.2 Automates de Büchi non-déterministes (NBA).

Avec ces NBA, on va définir une classe de langages équivalents aux expressions ω -régulières.

Définition 8.3. Un *NBA* est de la forme

$$\mathcal{A} = (Q, \Sigma, \delta, Q_0, F),$$

où

- ▷ Q est un ensemble *fini* d'états;
- ▷ Σ est l'alphabet d'entrée (non vide et fini);
- ▷ $\delta : Q \times \Sigma \rightarrow \wp(Q)$ est la *fonction de transition* (non-déterministe);
- ▷ $Q_0 \subseteq Q$ est l'ensemble des *états initiaux*;
- ▷ $F \subseteq Q$ est l'ensemble des *états acceptant*.

C'est la même définition que les automates finis non-déterministes (NFA) mais la différence est que l'on va exécuter ces automates sur des ω -mots, et on va donc modifier la définition d'*acceptation*.)

Définition 8.4. Une *exécution* de \mathcal{A} sur $\sigma \in \Sigma^\omega$ est un $\rho \in Q^\omega$ tel que

$$\rho(0) \in Q_0 \quad \text{et} \quad \forall n \in \mathbb{N} \quad \rho(n+1) \in \delta(\rho(n), \sigma(n)).$$

Remarque 8.1. L'ensemble

$$\{\rho \in Q^\omega \mid \rho \text{ est une exécution de } \mathcal{A} \text{ sur } \sigma\}$$

peut être non-dénombrable (par non déterminisme), et peut être vide.

¹Il y a plusieurs grammaires pour décrire ces expressions mais elles sont toutes (généralement) équivalentes.

– 38/59 –

Définition 8.5. Une exécution ρ de \mathcal{A} sur σ est acceptante si

$$\exists^\infty t, \quad \rho(t) \in F,$$

on visite infiniment souvent un état acceptant.

Remarque 8.2. Par le principe des tiroirs infinis (l'ensemble Q est fini), une exécution ρ est acceptante ssi il existe un état acceptant $q \in F$ que l'on visite une infinité de fois :

$$\exists q \in F, \quad \exists^\infty t \quad \rho(t) = q.$$

Définition 8.6. Le langage $\mathcal{L}_\omega(\mathcal{A})^2 \subseteq \Sigma^\omega$ est défini par

$$\mathcal{L}_\omega(\mathcal{A}) := \left\{ \sigma \in \Sigma^\omega \mid \exists \rho \in Q^\omega, \begin{array}{l} \rho \text{ est une exécution} \\ \text{acceptante de } \mathcal{A} \text{ sur } \sigma \end{array} \right\}.$$

Remarque 8.3. La fonction caractéristique de $\mathcal{L}_\omega(\mathcal{A})$:

$$\begin{aligned} \chi_{\mathcal{L}_\omega(\mathcal{A})} : \Sigma^\omega &\longrightarrow \{0, 1\} \\ \sigma &\longmapsto \begin{cases} 1 & \text{si } \sigma \in \mathcal{L}_\omega(\mathcal{A}) \\ 0 & \text{sinon.} \end{cases} \end{aligned}$$

Proposition 8.1. Étant donné un NBA \mathcal{A} , on peut tester si $\mathcal{L}_\omega(\mathcal{A}) = \emptyset$ en temps $O(|Q| + |\text{transitions}|)$.

Preuve. Il suffit de faire un DFS (c'est Tarjan-approved). \square

Exemple 8.3. Avec $\Sigma = \{a, b\}$, on considère le NBA \mathcal{P} défini comme

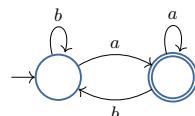


Fig. 8.1 | Automate de Büchi non-déterministe \mathcal{P}

où l'on note avec une flèche entrante les états initiaux et avec un double cercle les états acceptants. On a $\mathcal{L}_\omega(\mathcal{P}) = \mathcal{L}_\omega(P)$.

Exemple 8.4. On considère le NBA \mathcal{S} défini comme

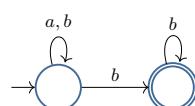


Fig. 8.2 | Automate de Büchi non-déterministe \mathcal{S}

On a $\mathcal{L}_\omega(\mathcal{S}) = \mathcal{L}_\omega(S)$.

²L'indice ω est très important ici.

Remarque 8.4. On note Π_2^0 l'ensemble des intersections dénombrables d'ouverts. On peut montrer que tout langage dans Π_2^0 peut s'écrire comme combinaison booléenne (union, intersection, complément) de langages comme $\mathcal{L}_\omega(P)$.

On note Σ_2^0 l'ensemble des unions dénombrables de fermés. On peut montrer que tout langage dans Σ_2^0 peut s'écrire comme combinaison booléenne (union, intersection, complément) de langages comme $\mathcal{L}_\omega(S)$.

Théorème 8.1. Soit Σ un alphabet non vide et fini.

1. Pour tout NBA \mathcal{A} sur Σ , il existe une expression ω -régulière G sur Σ telle que $\mathcal{L}_\omega(G) = \mathcal{L}_\omega(\mathcal{A})$.
2. Pour toute expression ω -régulière G sur Σ , il existe un NBA \mathcal{A} tel que $\mathcal{L}_\omega(\mathcal{A}) = \mathcal{L}_\omega(G)$.

De plus, ces deux « il existe » sont *constructifs*.

Preuve. 1. Soit $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ un NBA. Pour tout $p, q \in Q$, on note $\mathcal{A}_{p,q} = (Q, \Sigma, \delta, \{p\}, \{q\})$, et on a

$$\mathcal{L}_\omega(\mathcal{A}) = \bigcup_{p \in Q_0, q \in F} \mathcal{L}(\mathcal{A}_{p,q}) \cdot (\mathcal{L}(\mathcal{A}_{q,q}) \setminus \{\varepsilon\})^\omega.$$

2. On montre que :

- a) Soient \mathcal{A}, \mathcal{B} deux NBAs sur Σ . Il existe (constructif) un NBA \mathcal{C} sur Σ tel que

$$\mathcal{L}_\omega(\mathcal{C}) = \mathcal{L}_\omega(\mathcal{A}) \cup \mathcal{L}_\omega(\mathcal{B}).$$

- b) Soit \mathcal{A} un NFA sur Σ tel que $\varepsilon \notin \mathcal{L}(\mathcal{A})$, il existe (constructif) un NBA \mathcal{B} sur Σ tel que

$$\mathcal{L}_\omega(\mathcal{B}) = (\mathcal{L}(\mathcal{A}))^\omega.$$

- c) Soit \mathcal{A} un NFA sur Σ et \mathcal{B} un NBA sur Σ , il existe \mathcal{C} un NBA sur Σ tel que $\mathcal{L}_\omega(\mathcal{C}) = \mathcal{L}_\omega(\mathcal{A}) \cdot \mathcal{L}_\omega(\mathcal{B})$.

On ne détaillera pas toute la preuve.

- a) Il suffit de mettre les deux automates en parallèle (comme pour l'union disjointe de deux NFAs).

- b) Soit $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ un NFA tel que $\varepsilon \notin \mathcal{L}(\mathcal{A})$. On peut supposer qu'il n'y a pas de transition vers Q_0 et que $Q_0 \cap F = \emptyset$.³ On définit

$$\mathcal{B} := (Q, \Sigma, \delta_\mathcal{B}, Q_0, Q_0).$$

où

$$\delta_\mathcal{B}(q, a) := \begin{cases} \delta(q, a) & \text{si } \delta(q, a) \cap F = \emptyset \\ \delta(q, a) \cup Q_0 & \text{sinon.} \end{cases}$$

(Ce deuxième cas est important car on ne sait pas toujours le découpage du mot avant d'avoir considéré le mot en entier, ce qui demande de continuer l'exécution dans \mathcal{A} .)

- c) Il suffit de « mettre en série » \mathcal{A} puis \mathcal{B} en connectant les acceptants de \mathcal{B} avec les finaux de \mathcal{A} (quitte à prendre \mathcal{B}_t).

□

Définition 8.7. Un langage $L \subseteq \Sigma^\omega$ est ω -régulier s'il existe un NBA \mathcal{A} tel que $L = \mathcal{L}_\omega(\mathcal{A})$.

Corollaire 8.1. Si $L, K \subseteq \Sigma^\omega$ deux langages ω -réguliers alors $L \cup K$ est ω -régulier.

Dans la suite, on va voir la stabilité par intersections finies, mais on ne va pas voir de preuve du théorème ci-dessous.

Théorème 8.2 (Büchi). Si $L \subseteq \Sigma^\omega$ est ω -régulier alors $\Sigma^\omega \setminus L$ est ω -régulier.

Preuve. On l'admet (il n'est pas nécessaire pour le lien avec les formules LTL). □

8.3 Automates de Büchi déterministes (DBA).

Définition 8.8. Un NBA $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ est déterministe si

- ▷ $|Q_0| \leq 1$
- ▷ $\forall q \in Q, \forall a \in \Sigma, |\delta(q, a)| \leq 1$.

Proposition 8.2. Il n'existe pas de DBA \mathcal{D} sur Σ tel que

$$\mathcal{L}_\omega(\mathcal{D}) = \mathcal{L}_\omega((a + b)^* \cdot b^\omega).$$

(C'est le langage du NBA \mathcal{S} vu précédemment.)

Preuve. Par l'absurde, soit \mathcal{D} un DBA tel que $\mathcal{L}_\omega(\mathcal{D}) = \mathcal{L}_\omega((a + b)^* \cdot b^\omega)$. On peut supposer \mathcal{D} complète :

$$|Q_0| = 1 \quad \text{et} \quad \forall q \in Q, \forall a \in \Sigma, |\delta(q, a)| = 1.$$

(Il suffit d'ajouter un état puits.)

On notera donc $\delta : Q \times \Sigma \rightarrow Q$ et $Q_0 = \{q_\ell\}$. On définit

$$\begin{aligned} \delta^* : \Sigma^* &\longrightarrow Q \\ \varepsilon &\longmapsto q_\ell \\ u \cdot a &\longmapsto \delta(\delta^*(u), a). \end{aligned}$$

On a que $b^\omega \in \mathcal{L}_\omega(\mathcal{D})$, il existe donc $n_0 \geq 0$ tel que $\delta^*(b^{n_0}) \in F$. Puis, on a que $b^{n_0}ab^\omega \in \mathcal{L}_\omega(\mathcal{D})$, il existe donc $n_1 \geq 0$ tel que $\delta^*(b^{n_0}ab^{n_1}) \in F$. En itérant, il existe $(n_i)_{i \in \mathbb{N}}$ tel que

$$\forall i \in \mathbb{N}, \quad \delta^*(b^{n_0}ab^{n_1}ab^{n_2}a \cdots ab^{n_i}) \in F.$$

Considérons $\sigma = b^{n_0}ab^{n_1}ab^{n_2}a \cdots \in \Sigma^\omega$. Comme \mathcal{D} est déterministe, on a que $\sigma \in \mathcal{L}_\omega(\mathcal{D})$, ce qui est absurde. □

On ne peut donc pas déterminiser un automate de Büchi (ce qui explique que le théorème de Büchi n'est pas aussi simple que pour les NFAs).

Il existe plusieurs familles d'automates déterministes sur les ω -mots équivalents aux NBAs.

Pour faire un DBA équivalent à \mathcal{S} , on doit changer le formalisme.

³On peut prendre $\mathcal{A}_\ell := (Q \sqcup \{\ell\}, \Sigma, \delta_\ell, \{\ell\}, F)$ avec $\delta_\ell(\ell, a) = \bigcup_{q \in Q_0} \delta(q, a)$ et $\delta_\ell(q, a) = \delta(q, a)$ pour $q \neq \ell$. On a $\mathcal{L}(\mathcal{A}_\ell) = \mathcal{L}(\mathcal{A})$ car $\varepsilon \notin \mathcal{A}$.

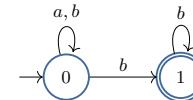


Fig. 8.3 | Automate de Büchi non-déterministe \mathcal{S}

On peut par exemple considérer le DBA suivant et dire que l'on ne peut voir l'état 1 au plus un nombre fini de fois.

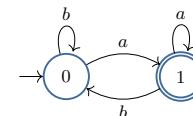


Fig. 8.4 | Automate de Büchi non-déterministe $\bar{\mathcal{S}}$

Il existe plusieurs familles de conditions d'acceptation : Muller, Rabin, Street, Parité, etc. Par exemple, pour Muller, un DBA est défini comme $(Q, \Sigma, \delta, q_i, \mathcal{T})$ où δ est déterministe et $\mathcal{T} \subseteq \wp(Q)$, et où $\rho \in Q^\omega$ est acceptantssi $\{q \mid \exists^\infty t, \rho(t) = q\} \in \mathcal{T}$.

8.4 Automates de Büchi généralisés (GNBA).

L'objectif des GNBA est pour la stabilité par intersection finies des expression ω -régulières, et seront utiles pour passer de LTL à un NBA.

Définition 8.9. Un GNBA est de la forme

$$\mathcal{G} = (Q, \Sigma, \delta, Q_0, \mathcal{F}),$$

où Q, Σ, δ, Q_0 sont définis comme pour les NBAs et où $\mathcal{F} \subseteq \wp(Q)$.

Une exécution de \mathcal{G} sur $\sigma \in \Sigma^\omega$ est acceptante si

$$\forall F \in \mathcal{F}, \quad \exists^\infty t \quad \rho(t) \in F.$$

Le langage $\mathcal{L}_\omega(\mathcal{G})$ est

$$\left\{ \sigma \in \Sigma^\omega \mid \begin{array}{l} \text{il existe une exécution} \\ \text{acceptante de } \mathcal{G} \text{ sur } \sigma \end{array} \right\}.$$

Remarque 8.5. Soit $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ un NBA alors $\mathcal{G} = (Q, \Sigma, \delta, Q_0, \{F\})$ est un GNBA tel que $\mathcal{L}_\omega(\mathcal{G}) = \mathcal{L}_\omega(\mathcal{A})$.

Lemme 8.1. Soient $\mathcal{G}_1, \mathcal{G}_2$ deux GNBAst avec $\mathcal{G}_i = (Q_i, \Sigma, \delta_i, Q_{0,i}, \mathcal{F}_i)$. Alors, $\mathcal{L}_\omega(\mathcal{G}_1) \cap \mathcal{L}_\omega(\mathcal{G}_2) = \mathcal{L}_\omega(\mathcal{G})$ où

$$\mathcal{G} = (Q_1 \times Q_2, \delta, Q_{0,1} \times Q_{0,2}, \mathcal{F})$$

avec

$$\delta((q_1, q_2), a) = \left\{ (q'_1, q'_2) \mid \begin{array}{l} q'_1 \in \delta_1(q_1, a) \\ q'_2 \in \delta_2(q_2, a) \end{array} \right\};$$

$$\mathcal{F} = \{F \times Q_2 \mid F \in \mathcal{F}_1\} \cup \{Q_1 \times F \mid F \in \mathcal{F}_2\}.$$

□

Théorème 8.3. Soit $\mathcal{G} = (Q, \Sigma, \delta, Q_0, \mathcal{F})$ où $\mathcal{F} = \{F_0, \dots, F_{k-1}\}$. Alors $\mathcal{L}_\omega(\mathcal{G}) = \mathcal{L}_\omega(\mathcal{A})$ où

$$\mathcal{A} = (Q \times \{0, \dots, k-1\}, \Sigma, \delta_{\mathcal{A}}, Q_0 \times \{0\}, F_0 \times \{0\}),$$

avec

$$\delta_{\mathcal{A}}((q, i), a) = \begin{cases} \{(q', i) \mid q' \in \delta(q, a)\} & \text{si } q \notin F_i \\ \{(q', i+1 \bmod k) \mid q' \in \delta(q, a)\} & \text{sinon} \end{cases}.$$

Preuve. ▷ Montrons $\mathcal{L}_\omega(\mathcal{A}) \subseteq \mathcal{L}_\omega(\mathcal{G})$. Soit $\sigma \in \mathcal{L}_\omega(\mathcal{A})$ et ρ une exécution acceptante de σ par \mathcal{A} . Soit $\rho \in Q^\omega$ une exécution de \mathcal{G} sur σ telle que $\rho = \pi_1 \circ \rho'$. Soit $t < t'$ tels que $\rho'(t) = \rho'(t') \in F_0 \times \{0\}$. Alors, $\rho'(t+1) \in Q \times \{1\}$ donc, il existe t_1, \dots, t_{k-1} tels que $\rho'(t_i) \in F_i \times \{i\}$ et $t < t_1 < t_2 < \dots < t_{k-1} < t'$. Comme il existe une infinité de t tels que $\rho(t) \in F_0 \times \{0\}$, on voit que ρ est acceptante.

▷ Montrons $\mathcal{L}_\omega(\mathcal{G}) \subseteq \mathcal{L}_\omega(\mathcal{A})$. Soit $\sigma \in \mathcal{L}_\omega(\mathcal{G})$ et $\rho \in Q^\omega$ une exécution de σ acceptante. Soit ρ' une exécution de \mathcal{A} sur σ telle que $\pi_1 \circ \rho' = \rho$. Supposons, par l'absurde, que pour un $i \in \{0, \dots, k-1\}$ tel que $\forall^\infty t, \pi_2(\rho(t)) = i$. Alors $\forall^\infty t, \pi_2(\rho'(t)) \notin F_i$. C'est impossible car $\exists^\infty t, \rho(t) \in F_i$. Il s'en suit que ρ' est acceptante.

□

Corollaire 8.2. Si L et K sont deux langages ω -réguliers alors $L \cap K$ est ω -régulier.

8.5 Traduction de LTL en (G)NBA.

Soit AP *fini* et ϕ une formule LTL sur AP. On veut montrer que $\llbracket \phi \rrbracket \subseteq (\mathbf{2}^{\text{AP}})^\omega$ est ω -régulier. Pour cela, on va construire un GNBA \mathcal{G} tel que $\mathcal{L}_\omega(\mathcal{G}) = \llbracket \phi \rrbracket$.

Remarque 8.6 (Idée). Soit $\sigma \in (\mathbf{2}^{\text{AP}})^\omega$. On « voudrait » \mathcal{G} tel que les exécutions ρ de \mathcal{G} sur σ sont de la forme

$$i \mapsto \{ \psi \mid \sigma \upharpoonright i \in \llbracket \psi \rrbracket \}.$$

Ceci n'est pas possible car on aurait une infinité d'états pour \mathcal{G} , ce que l'on ne peut pas avoir.

On suppose que LTL est défini par

$$\psi ::= a \mid \text{True} \mid \psi \wedge \psi \mid \neg \psi \mid \circ \psi \mid \psi \text{ Until } \psi$$

(on retire les redondances du LTL « habituel »).

Définition 8.10. On pose

$$\text{cl}(\phi) = \{ \psi, \neg \psi \mid \psi \text{ sous formule de } \phi \} / \psi \text{ identifié avec } \neg \neg \psi.$$

Les états de \mathcal{G} vont être certains $B \subseteq \text{cl}(\phi)$.

Définition 8.11. On dit que $B \subseteq \text{cl}(\phi)$ est maximal-consistant si :

- ▷ Si $\psi_1 \wedge \psi_2 \in \text{cl}(\phi)$ alors $\psi_1 \wedge \psi_2 \in B$ ssi $\psi_1, \psi_2 \in B$.
- ▷ Si $\text{True} \in \text{cl}(\phi)$ alors $\text{True} \in B$.
- ▷ Si $\psi \in \text{cl}(\phi)$, alors $\neg \psi \in B$ ssi $\psi \notin B$.

On dit que $B \subseteq \text{cl}(\phi)$ est localement consistant pour Until si

$$- 43/59 -$$

- ▷ $\psi_2 \in B$ implique $\psi_1 \text{ Until } \psi_2 \in B$;
- ▷ $\psi_1 \text{ Until } \psi_2 \in B$ et $\psi_2 \notin B$ implique $\psi_1 \in B$.

Définition 8.12. On pose

$$\mathcal{G} = (Q, \mathbf{2}^{\text{AP}}, \delta, Q_0, \mathcal{F}),$$

où

- ▷ $Q = \left\{ B \subseteq \text{cl}(\phi) \mid \begin{array}{l} B \text{ maximal-consistant} \\ B \text{ localement consistant pour Until} \end{array} \right\};$
- ▷ $Q_0 = \{B \in Q \mid \phi \in B\};$
- ▷ $\delta : Q \times \mathbf{2}^{\text{AP}} \rightarrow \wp(Q)$ est tel que $B' \in \delta(B, a)$ avec
 1. $B \cap \text{AP} = A \cap \text{cl}(\phi)$,
 2. si $\circ \psi \in \text{cl}(\phi)$ alors $\circ \psi \in B \iff \psi \in B'$,
 3. si $\psi_1 \text{ Until } \psi_2 \in \text{cl}(\phi)$ alors

$$\psi_1 \text{ Until } \psi_2 \in B \iff \left(\begin{array}{c} \psi_2 \in B \\ \text{ou} \\ \psi_1 \in B \text{ et } \psi_1 \text{ Until } \psi_2 \in B' \end{array} \right);$$

- ▷ $\mathcal{F} = \{F_{\psi_1 \text{ Until } \psi_2} \mid \psi_1 \text{ Until } \psi_2 \in \text{cl}(\phi)\}$ où

$$F_{\psi_1 \text{ Until } \psi_2} = \{B \in Q \mid \psi_1 \text{ Until } \psi_2 \in B \implies \psi_2 \in B\}.$$

Théorème 8.4. Avec cette définition, on a $\mathcal{L}_\omega(\mathcal{G}) = \llbracket \phi \rrbracket$.

Preuve. ▷ Montrons que $\llbracket \phi \rrbracket \subseteq \mathcal{L}_\omega(\mathcal{G})$. Soit $\sigma \in \llbracket \phi \rrbracket$ un mot. On définit $\rho \in \wp(\text{cl}(\phi))^\omega$ telle que

$$\rho(i) = \{ \psi \in \text{cl}(\phi) \mid \sigma \upharpoonright i \in \llbracket \psi \rrbracket \}.$$

On voit que, pour tout $i \in \mathbb{N}$, $\rho(i) \in Q$. On a que $\rho(0) \in Q_0$ car $\sigma \upharpoonright 0 = \sigma \in \llbracket \phi \rrbracket$. De plus, on a que $\rho(i+1) \in \delta(\rho(i), \sigma(i))$ (on vérifie aisément les trois propriétés sur δ). On montre que $\rho \in Q^\omega$ est acceptante. Soit $\psi_1 \text{ Until } \psi_2 \in \text{cl}(\phi)$. Si $\forall^\infty t, \sigma \upharpoonright t \notin \llbracket \psi_1 \text{ Until } \psi_2 \rrbracket$, alors $\forall^\infty t, (\psi_1 \text{ Until } \psi_2) \notin \rho(t)$. Sinon, $\exists^\infty t, \sigma \upharpoonright t \in \llbracket \psi_1 \text{ Until } \psi_2 \rrbracket$. Soit $N \in \mathbb{N}$. Il existe un certain $t \geq N$ tel que $\sigma \upharpoonright t \in \llbracket \psi_1 \text{ Until } \psi_2 \rrbracket$. Ainsi, il existe i tel que $\sigma \upharpoonright (t+i) \in \llbracket \psi_2 \rrbracket$ et pour tout $j \in \{0, \dots, i-1\}$, $\sigma \upharpoonright (t+j) \in \llbracket \psi_1 \rrbracket$. Donc $\psi_2, \psi_1 \text{ Until } \psi_2 \in \rho(t+i)$ et donc $\rho(t+i) \in F_{\psi_1 \text{ Until } \psi_2}$. On en conclut que ρ est acceptante.

- ▷ Montrons que $\mathcal{L}_\omega(\mathcal{G}) \subseteq \llbracket \phi \rrbracket$. Soit $\sigma \in \mathcal{L}_\omega(\mathcal{G})$ et ρ une exécution acceptante de \mathcal{G} sur σ . (On ne donne que l'idée pour cette partie de la preuve.) On va montrer que, pour tout $i \in \mathbb{N}$,

$$\rho(i) = \{ \psi \in \text{cl}(\phi) \mid \sigma \upharpoonright i \in \llbracket \psi \rrbracket \}.$$

On peut montrer par induction sur $\psi \in \text{cl}(\phi)$ que, pour tout $\rho \in Q^\omega$ et pour tout $\sigma \in (\mathbf{2}^{\text{AP}})^\omega$ tel que

- $\forall i, \rho(i+1) \in \delta(\rho(i), \sigma(i))$;
- $\forall F \in \mathcal{F}, \exists^\infty t, \rho(t) \in F$;

alors $\psi \in \rho(0) \iff \sigma \in \llbracket \psi \rrbracket$.

□

Remarque 8.7. Le GNFA \mathcal{G} a $O(2^{|\phi|})$ états et $|\mathcal{F}| = O(|\phi|)$. Donc, on a un NBA \mathcal{A} tel que $\mathcal{L}_\omega(\mathcal{A}) = \llbracket \phi \rrbracket$ et $|Q_{\mathcal{A}}| = O(2^{|\phi|+\log|\phi|})$.

Remarque 8.8. Supposons que $|\text{AP}| \geq 1$. Pour $n \in \mathbb{N}$, le langage

$$L_n := \left\{ A_1 \dots A_n A_1 \dots A_n \sigma \mid \sigma \in (\mathbf{2}^{\text{AP}})^\omega \right\}$$

est la sémantique de la formule

$$\phi_n = \bigwedge_{a \in \text{AP}} \bigwedge_{1 \leq i \leq n} (\circ^i a \leftrightarrow \circ^{n+i} a),$$

et on a que $|\phi_n| = O(n^2)$. Or, tout NBA \mathcal{A} pour L_n a $|Q_{\mathcal{A}}| = O(2^n)$.

Remarque 8.9. Soit $|\text{AP}| \geq 1$ et $a \in \text{AP}$. Soit

$$L := \{\sigma \in (\mathbf{2}^{\text{AP}})^\omega \mid \forall i, a \in \sigma(2i)\}.$$

On peut construire un NBA \mathcal{A} tel que $L = \mathcal{L}_\omega(\mathcal{A})$.

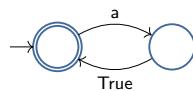


Fig. 8.5 | NBA sur l'alphabet $\mathbf{2}^{\text{AP}}$

Le langage L est le plus grand point fixe de $X = a \wedge \circ \circ X$ qui n'est pas définissable en LTL (le X doit être sous exactement un \circ pour être définissable dans LTL). Il n'existe donc pas de formule ϕ LTL telle que $\llbracket \phi \rrbracket = L$.

8.6 Théorème de Büchi : complémentation pour les langages ω -réguliers.

Soit Σ un alphabet fini. L'objectif de cette section est de montrer le théorème suivant.

Théorème 8.2 (Büchi). Si $L \subseteq \Sigma^\omega$ est ω -régulier alors $\Sigma^\omega \setminus L$ est ω -régulier.

Notation. Pour $\sigma \in \Sigma^\omega$ et $n \leq k$, on note $\sigma[n : k] = \sigma(n) \dots \sigma(k-1)$ avec la convention $\sigma[n : n] = \varepsilon$.

Notation. Pour $\mathcal{V} \in \Sigma^*$, on note \mathcal{V}^ω est l'ensemble des $\sigma \in \Sigma^\omega$ tels qu'il existe $(k_n)_{n \in \mathbb{N}}$ strictement croissante avec $\sigma[k_n : k_{n+1}] \in \mathcal{V}$ pour tout $n \in \mathbb{N}$.

Soit $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ un NBA. On va construire une expression ω -régulière dont le langage est $\Sigma^\omega \setminus \mathcal{L}_\omega(\mathcal{A})$.

Notation. Soient $q, q' \in Q$ $\omega \in \Sigma^*$. On note

- ▷ $q \xrightarrow{w} q'$ si $w \in \mathcal{L}(\mathcal{A}_{q,q'})$;
- ▷ $q \xrightarrow{w}_F q'$ si $q \xrightarrow{w} q'$ en passant par F (possiblement en q et/ou en q').

Définition 8.13. On définit la relation $\sim_{\mathcal{A}}$ sur Σ^* par :

$$u \sim_{\mathcal{A}} v \iff \forall q, q' \in Q, \begin{cases} q \xrightarrow{u} q' \iff q \xrightarrow{v} q' \\ \text{ET} \\ q \xrightarrow{u}_F q' \iff q \xrightarrow{v}_F q' \end{cases}.$$

Lemme 8.2. La relation $\sim_{\mathcal{A}}$ est une congruence. C'est-à-dire, une relation d'équivalence qui vérifie

$$u \sim_{\mathcal{A}} u' \implies vuw \sim_{\mathcal{A}} vu'w.$$

Lemme 8.3. La congruence $\sim_{\mathcal{A}}$ est d'*index fini* (i.e. a un nombre fini de classes d'équivalences). Ses classes d'équivalences $\mathcal{V} \subseteq \Sigma^*$ sont des langages réguliers.

Preuve. (Idée) La classe d'équivalence de $u \in \Sigma^*$ est exactement

$$\begin{aligned} & \bigcap_{q, q' \in Q} \{S(q, q') \mid u \in S(q, q')\} \\ & \cap \bigcap_{q, q' \in Q} \{S^F(q, q') \mid u \in S^F(q, q')\} \\ & \cap \bigcap_{q, q' \notin Q} \{S(q, q') \mid u \notin S(q, q')\} \\ & \cap \bigcap_{q, q' \notin Q} \{S^F(q, q') \mid u \notin S^F(q, q')\} \end{aligned},$$

où l'on note

- ▷ $S(q, q') = \{w \in \Sigma^* \mid q \xrightarrow{w} q'\}$;
- ▷ $S^F(q, q') = \{w \in \Sigma^* \mid q \xrightarrow{w}_F q'\}$;

□

Lemme 8.4. La congruence $\sim_{\mathcal{A}}$ *sature* $\mathcal{L}_\omega(\mathcal{A})$: si $U, V \subseteq \Sigma^*$ sont des classes d'équivalences telles que $UV^\omega \cap \mathcal{L}_\omega(\mathcal{A}) \neq \emptyset$ alors $UV^\omega \subseteq \mathcal{L}_\omega(\mathcal{A})$.

Preuve. (Idée) Soit $\sigma \in UV^\omega \cap \mathcal{L}_\omega(\mathcal{A})$. Soit ρ une exécution acceptante de \mathcal{A} sur σ . Soit $(k_n)_{n \in \mathbb{N}}$ strictement croissante telle que $\sigma[0 : k_0] \in U$ et $\sigma[k_n : k_{n+1}] \in V$ pour tout $n \geq 0$.

On remarque que $\sigma[k_n : k_{n+1}] \in S(\rho(k_n), \rho(k_{n+1}))$ et il existe une infinité de n tels que $\sigma[k_n : k_{n+1}] \in S^F(\rho(k_n), \rho(k_{n+1}))$.

Soit $\beta \in UV^\omega$ et soit $(\ell_n)_{n \in \mathbb{N}}$ strictement croissante telle que $\beta[0 : \ell_0] \in U$ et $\beta[\ell_n : \ell_{n+1}] \in V$ pour tout $n \in \mathbb{N}$. Donc $\beta[0 : \ell_0] \sim_{\mathcal{A}} \sigma[0 : k_0] \in S(\rho(0), \rho(k_0))$ et $\beta[\ell_n : \ell_{n+1}] \sim_{\mathcal{A}} \sigma[k_n : k_{n+1}]$. En particulier, on a $\beta[\ell_n : \ell_{n+1}] \in S(\rho(k_n), \rho(k_{n+1}))$ et il existe une infinité de n tels que $\beta[\ell_n : \ell_{n+1}] \in S^F(\rho(k_n), \rho(k_{n+1}))$. □

Proposition 8.3. Soit \sim une congruence d'*index fini* sur Σ^* . Si $\sigma \in \Sigma^\omega$ alors il existe $U, V \subseteq \Sigma^*$ des classes d'équivalences telles que $\sigma \in UV^\omega$.

Preuve. Fixons \sim et $\sigma \in \Sigma^\omega$. Pour $k, \ell \in \mathbb{N}$ et $n > k, \ell$, on note $k \equiv_\sigma^n \ell$ ssi $\sigma[k : n] \sim \sigma[\ell : n]$. Puis, on note $k \equiv_\sigma \ell$ s'il existe un certain $n > k, \ell$ tel que $k \equiv_\sigma^n \ell$.

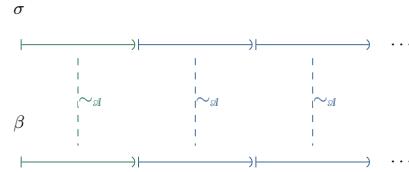
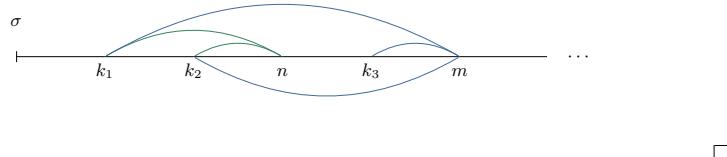


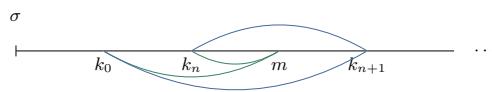
Fig. 8.6 | Visuel de la preuve du lemme 8.4

Lemme 8.5. La relation \equiv_σ est une relation d'équivalence d'index fini.

Preuve. Pour la transitivité, si $k_1 \equiv_\sigma k_2$ et $k_2 \equiv_\sigma k_3$, on peut regarder la figure ci-dessous.



La relation \equiv_σ est d'index fini donc par le principe des tiroirs infinis, il existe $(k_n)_{n \in \mathbb{N}}$ strictement croissante telle que $k_n \equiv_\sigma k_m$ pour toute paire n, m . Quitte à prendre une sous séquence, on peut supposer que, pour $n \geq 1$, on a $\sigma[k_0 : k_n] \sim_\sigma \sigma[k_n : k_{n+1}]$.



Par le principe des tiroirs infinis, on peut supposer, pour tout $n, m \geq 1$ que $\sigma[k_0 : k_n] \sim \sigma[k_0 : k_m]$ et on a donc, pour tout $n, m \geq 0$, on ait $\sigma[k_n : k_{n+1}] \sim \sigma[k_m : k_{m+1}]$. Il suffit donc de prendre U la classe d'équivalence de $\sigma[0 : k_0]$ et V la classe des $\sigma[k_n : k_{n+1}]$. □

Remarque 8.10. Dans la preuve, on a $VV \subseteq V$, i.e. si $u, v \in V$ alors $uv \in V$.

Corollaire 8.3. Soit \sim une congruence d'index fini sur Σ^* et soit $L \subseteq \Sigma^\omega$ saturé par \sim . Alors

$$L = \bigcup \left\{ UV^\omega \mid \begin{array}{l} UV^\omega \cap L \neq \emptyset \\ U, V \text{ classes pour } \sim \end{array} \right\}.$$

Théorème 8.2 (Büchi). Si $L \subseteq \Sigma^\omega$ est ω -régulier alors $\Sigma^\omega \setminus L$ est ω -régulier.

Preuve. Avec $L = \mathcal{L}_\omega(\mathcal{A})$, on a que $\Sigma^\omega \setminus L$ est saturé par $\sim_\mathcal{A}$. Si U et V sont des classes d'équivalences pour $\sim_\mathcal{A}$, alors $UV^\omega \cap (\Sigma^\omega \setminus L) \neq \emptyset$ et donc $UV^\omega \cap L = \emptyset$ donc $UV^\omega \subseteq \Sigma^\omega \setminus L$. Donc,

$$\Sigma^\omega \setminus L = \bigcup \left\{ UV^\omega \mid \begin{array}{l} UV^\omega \cap L = \emptyset \\ U, V \text{ classes pour } \sim \end{array} \right\},$$

et donc $\Sigma^\omega \setminus L$ est ω -régulier car

▷ $U, V \subseteq \Sigma^\omega$ sont des $\sim_\mathcal{A}$ -classes donc réguliers ;

▷ $\sim_\mathcal{A}$ est d'index fini.

De plus, on peut décider si $UV^\omega \cap L = \emptyset$ donc on peut construire une expression ω -régulière. □

IX. Bisimulation.

L'idée de ce chapitre est d'identifier les systèmes de transitions avec une « même structure de branchement ». Par exemple, on identifie les deux systèmes suivants (le premier est un « dépliement » du second).

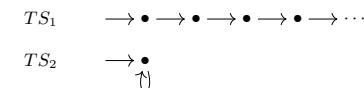


Fig. 9.1 | Deux systèmes de transitions identifiés par bisimulation

Définition 9.1. Soient TS_0 et TS_1 où

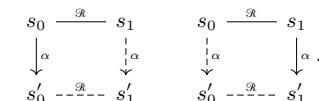
$$TS_i = (S_i, \text{Act}, \rightarrow_i, I_i, \text{AP}, L_i),$$

où l'on note $s_i \xrightarrow{\alpha} s'_i$ où $\alpha \in \text{Act}$ et $s_i, s'_i \in S_i$, et $L_i : S_i \rightarrow \wp(\text{AP})$.

Une *bisimulation* entre TS_0 et TS_1 est une relation $\mathcal{R} \subseteq S_0 \times S_1$ telle que

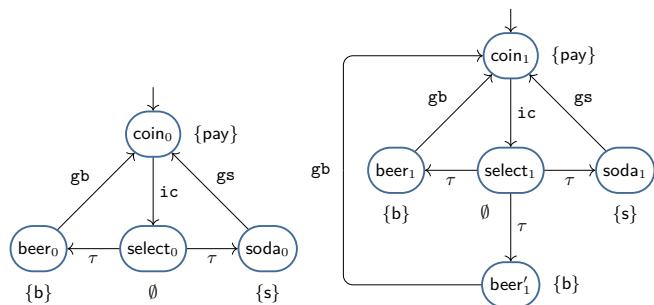
1. si $s_0 \mathcal{R} s_1$ alors $L_0(s_0) = L_1(s_1)$;
2. si $s_0 \mathcal{R} s_1$ et $s_0 \xrightarrow{\alpha} s'_0$, alors il existe $s'_1 \in S_1$ tel que $s_1 \xrightarrow{\alpha} s'_1$ et $s'_0 \mathcal{R} s'_1$.
3. de même symétriquement.

Ces deux dernières conditions peuvent être visualisées comme les deux diagrammes ci-dessous.



Exemple 9.1. Avec les deux systèmes de transitions suivants, on peut construire une bisimulation \mathcal{R} avec

$$\mathcal{R} = \left\{ \begin{array}{l} (\text{coin}_0, \text{coin}_1), (\text{select}_0, \text{select}_1), (\text{beer}_0, \text{beer}_1), \\ (\text{beer}_0, \text{beer}'_1), (\text{soda}_0, \text{soda}_1) \end{array} \right\}.$$



Définition 9.2. Soient TS_0 et TS_1 deux systèmes de transitions. La relation de *bisimilarité* entre TS_0 et TS_1 est donnée par

$$s_0 \sim s_1 \iff \exists \mathcal{R} \text{ une bisimulation, } s_0 \mathcal{R} s_1.$$

On dit alors que s_0 est *bisimilaire* à s_1 .

Lemme 9.1. 1. Étant donné TS , on a que s est bisimilaire à s (car $\{(s, s) \mid s \in S\}$ est une bisimulation entre TS et lui-même).

2. Si \mathcal{R} est une bisimulation entre TS_1 et TS_2 , alors

$$\mathcal{R}^{-1} := \{(s_2, s_1) \mid (s_1, s_2) \in \mathcal{R}\}$$

est une bisimulation entre TS_2 et TS_1 .

3. Si \mathcal{R} est une bisimulation entre TS_1 et TS_2 , et \mathcal{S} est une bisimulation entre TS_2 et TS_3 , alors

$$\mathcal{S} \circ \mathcal{R} := \left\{ (s_1, s_3) \mid \exists s_2, \begin{array}{l} (s_1, s_2) \in \mathcal{R} \\ (s_2, s_3) \in \mathcal{S} \end{array} \right\}$$

est une bisimulation entre TS_1 et TS_3 . \square

Lemme 9.2. 1. La relation de bisimilarité \sim entre TS_1 et TS_2 est une bisimulation.

2. Si \mathcal{R} est une bisimulation entre TS_1 et TS_2 alors $\mathcal{R} \subseteq \sim$.

3. La relation \sim entre TS et lui-même est une relation d'équivalence.

Preuve. (Idée)

1. Soient $s_1 \sim s_2$, et soit donc \mathcal{R} telle que $s_1 \mathcal{R} s_2$. Donc,

▷ $L_1(s_1) = L_2(s_2)$;

▷ comme $s'_1 \mathcal{R} s'_2$, on a $s'_1 \sim s'_2$ où

$$\begin{array}{ccc} s_0 & \xrightarrow{\mathcal{R}} & s_1 \\ \downarrow \alpha & & \downarrow \alpha \\ s'_0 & \dashv \mathcal{R} \dashv & s'_1 \end{array}$$

▷ et symétriquement pour le cas dual.

2. Si $s_1 \mathcal{R} s_2$ alors $s_1 \sim s_2$ par définition de \sim .

3. On a les points suivants.

Réflexivité. On a $s \sim s$.

Symétrie. Si $s \sim s'$ alors $s \mathcal{R} s'$ pour une certaine bisimulation \mathcal{R} , et donc $s' \mathcal{R}^{-1} s$ et donc $s' \sim s$.

Transitivité. Si $s \sim s'$ et $s' \sim s''$ alors $s \mathcal{R} s'$ et $s' \mathcal{R} s''$, et donc $s (\mathcal{S} \circ \mathcal{R}) s''$ d'où $s \sim s''$. \square

9.1 Quotient par bisimulation.

Définition 9.3 (Quotient par bisimulation). Soit un système de transitions $TS = (S, \text{Act}, \rightarrow, I, \text{AP})$, On définit TS_\sim par :

- ▷ $S_\sim := \{[s]_\sim \mid s \in S\}$;
- ▷ $\text{Act}_\sim := \text{Act}$;
- ▷ $[s]_\sim \xrightarrow{\alpha} [s']_\sim$ ssi $s \xrightarrow{\alpha} s'$;
- ▷ $I_\sim = \{[s]_\sim \mid s \in I\}$;
- ▷ $\text{AP}_\sim := \text{AP}$;
- ▷ $L_\sim([s]_\sim) = L(s)$.

Lemme 9.3. Pour tout $s \in I$ et $s \sim [s]_\sim$.

Définition 9.4. Soient $TS_1 = (S_1, \text{Act}, \rightarrow_1, I_1, \text{AP}, L_1)$ et $TS_2 = (S_2, \text{Act}, \rightarrow_2, I_2, \text{AP}, L_2)$ deux systèmes de transitions.

On note $TS_1 \approx TS_2$ s'il existe une bisimulation \mathcal{R} entre TS_1 et TS_2 telle que

1. pour tout $s_1 \in I_1$, il existe $s_2 \in I_2$ tel que $s_1 \mathcal{R} s_2$;
2. pour tout $s_2 \in I_2$, il existe $s_1 \in I_1$ tel que $s_1 \mathcal{R} s_2$.

9.2 Bisimilarité et équivalence de traces.

Proposition 9.1. Soient TS_1, TS_2 sur un même ensemble d'actions Act et de propositions atomiques AP . Si $TS_1 \approx TS_2$ alors

$$\text{Tr}^\omega(TS_1) = \text{Tr}^\omega(TS_2).$$

Corollaire 9.1. Si $TS_1 \approx TS_2$ alors, pour toute propriété $\text{LT } P \subseteq (2^{\text{AP}})^\omega$, on a

$$TS_1 \models P \iff TS_2 \models P.$$

Exemple 9.2. Pour un système de transitions TS , on a $TS \approx TS_\sim$, on peut donc se ramener au quotient par bisimilarité, pour tester si $TS \models P$.

Exemple 9.3. Le quotient par bisimilarité du système de transitions

10.3 Équivalences logiques.

10.3.1 Équivalences logiques pour les formules.

Définition 10.5. Soient AP, Act, et soient ϕ, ψ des formules. On note $\phi \equiv \psi$ ssi

$$\forall M = (S, \text{Act}, \rightarrow, \text{AP}, L), \quad \llbracket \phi = \psi \rrbracket,$$

autrement dit $\forall M, \forall s \in S, \quad s \Vdash \phi \iff s \Vdash \psi$.

Lemme 10.1. On a que $[\alpha] -$ et $\langle \alpha \rangle -$ sont duals par De Morgan :

$$[\alpha]\phi \equiv \neg\langle\alpha\rangle\neg\phi \quad \text{et} \quad \langle\alpha\rangle\phi \equiv \neg[\alpha]\neg\phi.$$

De plus, comme $\langle\alpha\rangle -$ est défini comme quantification existentielle, on a

$$\langle\alpha\rangle(\phi \vee \psi) \equiv \langle\alpha\rangle\phi \vee \langle\alpha\rangle\psi \quad \text{et} \quad \langle\alpha\rangle\perp \equiv \perp.$$

De même, comme $[\alpha] -$ est défini comme quantification universelle, on a

$$[\alpha](\phi \wedge \psi) \equiv [\alpha]\phi \wedge [\alpha]\psi \quad \text{et} \quad [\alpha]\top \equiv \top.$$

10.3.2 Équivalences logiques pour les états.

Définition 10.6. Soient Act et AP. On se donne M_1 , et M_2 deux modèles de Kripke. Pour $s_1 \in S_1$ et $s_2 \in S_2$, on note

$$s_1 \equiv s_2 \quad \text{ssi} \quad \forall \phi, \quad s_1 \Vdash \phi \iff s_2 \Vdash \phi.$$

Remarque 10.1. On aurait pu définir une telle équivalence pour LML, mais on aurait que $\sigma_1 \equiv \sigma_2$ ssi $\sigma_1 = \sigma_2$, ce qui n'est pas très intéressant. Avec des modèles de Kripke et la logique HML, on a des états différents qui sont équivalents.

Théorème 10.1. Si $s_1 \sim s_2$ alors $s_1 \equiv s_2$.

Preuve. Par induction sur ϕ , on montre que

$$s_1 \sim s_2 \implies (s_1 \Vdash \phi \iff s_2 \Vdash \phi).$$

Cas de $a \in AP$. Si $s_1 \sim s_2$ alors $L_1(s_1) = L_2(s_2)$ donc $s_1 \Vdash a$ ssi $s_2 \Vdash a$.

Cas de $- \wedge -, - \vee -, \top, \perp, \text{et } \neg-$. On applique l'hypothèse d'induction et/ou l'hypothèse $s_1 \sim s_2$, et on conclut.

Cas de $\langle\alpha\rangle -$. Supposons $s_1 \sim s_2$. Supposons $s_1 \Vdash \langle\alpha\rangle\phi$. Il existe donc s'_1 tel que $s_1 \xrightarrow{\alpha} s'_1$ et $s'_1 \Vdash \phi$. Par bisimulation

$$\begin{array}{ccc} s_1 & \xrightarrow{\sim} & s_2 \\ \downarrow^{\alpha} & & \downarrow^{\alpha}, \\ s'_1 & \xrightarrow{\sim} & s'_2 \end{array}$$

il existe donc s'_2 tel que $s_2 \xrightarrow{\alpha} s'_2$ et $s'_1 \sim s'_2$. Donc, par hypothèse d'induction, on a que $s'_2 \Vdash \phi$. On a donc $s_2 \Vdash \langle\alpha\rangle\phi$. On procéderait de même dans l'autre cas, on a bien que

$$s_1 \Vdash \langle\alpha\rangle\phi \iff s_2 \Vdash \langle\alpha\rangle\phi.$$

Cas de $[\alpha] -$. On a que $[\alpha]\phi \equiv \neg\langle\alpha\rangle\neg\phi$.

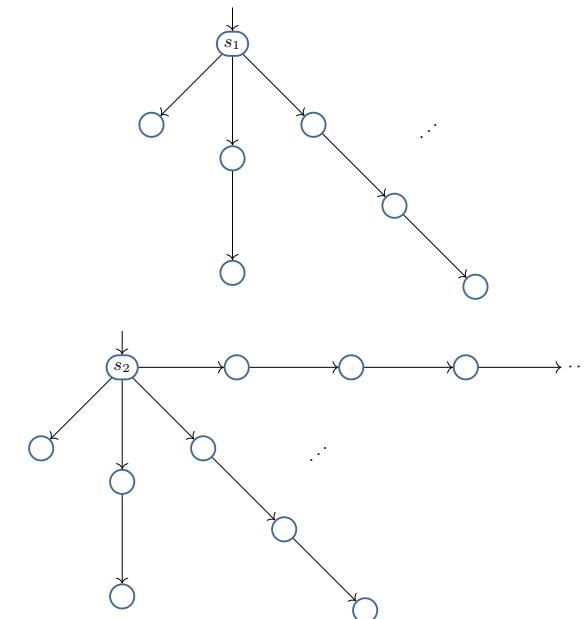
□

10.4 Propriété de Hennessy-Milner.

Remarque 10.2 (Question). Est-ce que $s_1 \equiv s_2$ implique $s_1 \sim s_2$? Autrement dit, est-ce que \equiv est une bisimulation?

La réponse est **non** en général.

Exemple 10.2. Considérons les systèmes de transitions suivants.



On remarque que $s_1 \not\sim s_2$ mais $s_1 \equiv s_2$ (la logique HML ne peut « voir » qu'à une profondeur finie).

Définition 10.7. Une classe \mathfrak{M} de modèles a la *propriété de Hennessy-Milner* si, dans \mathfrak{M} , l'équivalence \equiv entre états est une bisimulation.

Définition 10.8. Soit $K = (S, \text{Act}, \rightarrow)$ alors, pour $\alpha \in \text{Act}$ et $s \in S$, on note

$$\text{Succ}^\alpha(s) := \{s' \in S \mid s \xrightarrow{\alpha} s'\}.$$

Définition 10.9. Un modèle de Kripke M est dit à *image finie* si, pour tout $s \in S$ et $\alpha \in \text{Act}$, l'ensemble $\text{Succ}^\alpha(s)$ des successeurs de s par une action α fixée est finie.

Proposition 10.1 (Hennessy-Milner). La classe des modèles à image finie a la propriété de Hennessy-Milner.

10.5 Saturation modale.

Définition 10.10. Soit $M = (S, \text{Act}, \rightarrow, \text{AP}, L)$.

- Soient $T \subseteq S$ et $\Phi \subseteq \text{HML}$. On dit que Φ est *satisfiable* dans T s'il existe $s \in T$ tel que pour tout $\phi \in \Phi$, on a $s \Vdash \phi$.

2. Soient $T \subseteq S$ et $\Phi \subseteq \text{HML}$. On dit que Φ est *finiement satisfiable* dans T si, tout $\Psi \subseteq_{\text{fin}} \Phi$ est satisfiable dans T , autrement dit

$$\forall \Psi \subseteq_{\text{fin}} \Phi \quad \exists s \in T \quad s \Vdash \bigwedge \Psi.$$

3. On dit que M est *modalement saturée* si, pour tout $s \in S$, tout $\alpha \in \text{Act}$ et tout $\Phi \subseteq \text{HML}$ tel que Φ est finiement satisfiable dans $\text{Succ}^\alpha(s)$ alors Φ est satisfiable dans $\text{Succ}^\alpha(s)$, autrement dit

$$\begin{aligned} & \forall s \in S, \forall \alpha \in \text{Act}, \forall \Phi \subseteq \text{HML}, \\ & \left(\begin{array}{c} \forall \Psi \subseteq_{\text{fin}} \Phi \quad \exists s' \xleftarrow{\alpha} s \quad s' \Vdash \bigwedge \Psi \\ \Downarrow \\ \exists s' \xleftarrow{\alpha} s \quad \forall \phi \in \Phi \quad s' \Vdash \phi \end{array} \right). \end{aligned}$$

Proposition 10.2. Si M_1 et M_2 sont modalement saturés alors \equiv entre S_1 et S_2 est une bisimulation.

Preuve. Soient $s_1 \equiv s_2$.

1. On a bien $L_1(s_1) = L_2(s_2)$.
2. Supposons $s_1 \xrightarrow{\alpha} s'_1$. On pose $\Phi = \{\phi \mid s'_1 \Vdash \phi\}$. Soit $\Psi \subseteq_{\text{fin}} \Phi$ quelconque. On a $s_1 \Vdash \langle \alpha \rangle \bigwedge \Psi$ donc $s_2 \Vdash \langle \alpha \rangle \bigwedge \Psi$. Il existe donc $s'_2 \xleftarrow{\alpha} s_2$ telle que $s'_2 \Vdash \bigwedge \Psi$. Par **saturation modale** (de M_2), il existe $s'_2 \xleftarrow{\alpha} s_2$ tel que, pour tout $\phi \in \Phi$, $s'_2 \Vdash \phi$. Autrement dit, on a bien :

$$\begin{array}{ccc} s_1 & \xlongequal{\equiv} & s_2 \\ \downarrow^{\alpha} & & \downarrow^{\alpha} \\ s'_1 & \dashv\dashv & s'_2 \end{array}$$

3. Symétriquement.

□

Proposition 10.3. Si M est à image finie alors M est modalement saturé.

Preuve. Soit $s \in S$, $\alpha \in \text{Act}$, et $\Phi \subseteq \text{HML}$. Supposons que Φ finiement satisfiable dans $\text{Succ}^\alpha(s)$. Par l'absurde, supposons

$$\forall t \xrightarrow{\alpha} s \quad \exists \phi_t \in \Phi \quad t \not\Vdash \phi_t.$$

On pose

$$\Psi := \{\phi_t \mid t \in \text{Succ}^\alpha(s)\} \subseteq_{\text{fin}} \Phi.$$

Il existe donc $t \xleftarrow{\alpha} s$ tel que $t \Vdash \bigwedge \Psi$, donc $t \Vdash \phi_t$. **Absurde.**

□

10.6 Algèbres de Boole avec opérateurs.

Soit $\mathcal{L}(\text{HML}) = \{[\phi]_\equiv \mid \phi \in \text{HML}\}$. On notera dans la suite ϕ pour $[\phi]_\equiv$. Comme pour le *Homework*, on a que $(\mathcal{L}(\text{HML}), \leq)$ est une algèbre de Boole où $\phi \leq \psi$ ssi $\phi \wedge \psi \equiv \phi$. Pour un modèle de Kripke M , on a que $(\wp(M), \subseteq)$ est une algèbre de Boole et

$$\begin{aligned} \llbracket - \rrbracket : \mathcal{L}(\text{HML}) &\longrightarrow \wp(S) \\ [\phi]_\equiv &\longmapsto \llbracket \phi \rrbracket \end{aligned}$$

est un morphisme d'algèbres de Boole.

Lemme 10.2. Soit M un modèle de Kripke. On a que

$$\llbracket \langle \alpha \rangle \rrbracket : \wp(S) \longrightarrow \wp(S)$$

$$A \longmapsto \{s \mid \exists s' \xleftarrow{\alpha} s \quad s' \in A\}$$

est un morphisme de \vee -semi-treillis, et on a $\llbracket \langle \alpha \rangle \phi \rrbracket = \llbracket \langle \alpha \rangle \rrbracket(\llbracket \phi \rrbracket)$. De même, on a que

$$\llbracket [\alpha] \rrbracket : \wp(S) \longrightarrow \wp(S)$$

$$A \longmapsto \{s \mid \forall s' \xleftarrow{\alpha} s \quad s' \in A\}$$

est un morphisme de \wedge -semi-treillis, et on a $\llbracket [\alpha] \phi \rrbracket = \llbracket [\alpha] \rrbracket(\llbracket \phi \rrbracket)$. □

Lemme 10.3. On a que

$$\langle \alpha \rangle : \mathcal{L}(\text{HML}) \longrightarrow \mathcal{L}(\text{HML})$$

$$[\phi]_\equiv \longmapsto [\langle \alpha \rangle \phi]_\equiv$$

est un morphisme de \vee -semi-treillis et

$$[\alpha] : \mathcal{L}(\text{HML}) \longrightarrow \mathcal{L}(\text{HML})$$

$$[\phi]_\equiv \longmapsto [[\alpha] \phi]_\equiv$$

est un morphisme de \wedge -semi-treillis

On a $\llbracket [\alpha] \rrbracket = \llbracket \langle \alpha \rangle \rrbracket^\partial$ et $[\alpha] = \langle \alpha \rangle^\partial$ avec la notation définie ci-après.

Définition 10.11. Pour $f : B \rightarrow B'$ où B, B' sont deux algèbres de Boole, on note

$$\begin{aligned} f^\partial : B &\longrightarrow B' \\ a &\longmapsto \neg' f(\neg a). \end{aligned}$$

C'est le dual par De Morgan de f .

Lemme 10.4. Pour $f : B \rightarrow B'$ où B, B' sont deux algèbres de Boole,

1. on a $(f^\partial)^\partial = f$;
2. si f est un morphisme de \vee -semi-treillis alors f^∂ est un morphisme de \wedge -semi-treillis;
3. si f est un morphisme de \wedge -semi-treillis alors f^∂ est un morphisme de \vee -semi-treillis;
4. si f est un morphisme de treillis alors $f^\partial = f$.

Remarque 10.3. Dans LML , on a une modalité \circ qui est auto duale car morphisme de treillis.

Définition 10.12 (BAO). Une *algèbre de Boole avec opérateurs* (BAO) est de la forme

$$(B, \leq, (f_\alpha)_{\alpha \in \text{Act}})$$

où (B, \leq) est une algèbre de Boole et $f_\alpha : B \rightarrow B$ est un morphisme de \vee -semi-treillis pour tout $\alpha \in \text{Act}$.

Exemple 10.3. On a que $\mathcal{L}(\text{HML})^+ := (\mathcal{L}(\text{HML}), \leq, (\langle \alpha \rangle)_{\alpha \in \text{Act}})$ est une algèbre de Boole avec opérateurs.

Exemple 10.4. Pour $L = (S, \text{Act}, \rightarrow)$ une structure de Kripke, alors

$$K^+ := (\wp(S), \subseteq, (\llbracket \langle \alpha \rangle \rrbracket)_{\alpha \in \text{Act}})$$

est une algèbre de Boole avec opérateurs.

10.7 Structures ultrafiltres.

Définition 10.13. Soit $(B, \leq, (f_\alpha)_{\alpha \in \text{Act}})$ une algèbre de Boole avec opérateurs. On définit

$$\mathfrak{Uf}(B) := (\mathbf{Sp}(B), \text{Act}, \rightarrow),$$

où $\mathcal{F} \xrightarrow{\alpha} \mathcal{H}$ ssi pour tout $b \in B$, $b \in \mathcal{H}$ implique $f_\alpha(b) \in \mathcal{F}$.

Lemme 10.5. On a $\mathcal{F} \xrightarrow{\alpha} \mathcal{H}$ ssi pour tout $b \in B$, $f_\alpha^\partial(b) \in \mathcal{F}$ implique $b \in \mathcal{H}$.

Preuve. (Idée) Supposons $\mathcal{F} \xrightarrow{\alpha} \mathcal{H}$. Soit $b \in B$. Par contraposée, supposons $b \notin \mathcal{H}$. On a donc $\neg b \in \mathcal{H}$ et donc $f_\alpha(\neg b) \in \mathcal{F}$. Mais, $f_\alpha(\neg b) = \neg f_\alpha^\partial(b)$ et donc $f_\alpha^\partial(b) \notin \mathcal{F}$.

Réciproquement, supposons $\mathcal{F} \not\xrightarrow{\alpha} \mathcal{H}$. Il existe donc $b \in B$ tel que $b \in \mathcal{H}$ et $f_\alpha(b) \notin \mathcal{F}$. Donc $\neg f_\alpha(b) = f_\alpha^\partial(\neg b) \in \mathcal{F}$ et $\neg b \notin \mathcal{H}$. \square

10.7.1 Extensions d'ultrafiltres aux modèles de Kripke.

Soit $M = (S, \text{Act}, \rightarrow, \text{AP}, L)$.

Remarque 10.4. Soit X un ensemble.

- ▷ Un *filtre (propre)* sur X est un filtre (propre) sur $(\wp(X), \subseteq)$.
- ▷ Un *ultrafiltre* sur X est un ultrafiltre sur $(\wp(X), \subseteq)$.
- ▷ Si $G \subseteq \wp(X)$ qui a la propriété des intersections finies (*finite intersection property*) alors

$$\bigcap \{E \mid E \text{ filtre propre et } E \supseteq G\}$$

est un filtre propre.

- ▷ Si $G \subseteq \wp(X)$ a la propriété des intersections finies alors il existe un ultrafiltre \mathcal{F} sur X tel que $G \subseteq \mathcal{F}$.
- ▷ Pour tout $x \in X$, on définit

$$\pi(x) := \{A \in \wp(X) \mid x \in A\}$$

est l'*ultrafiltre principal* induit par x . On a que

$$\pi : X \rightarrow \mathfrak{Uf}(X) =: \mathbf{Sp}(\wp(X), \subseteq).$$

Remarque 10.5. La fonction de *labelling* $L : S \rightarrow 2^{\text{AP}}$ peut être décrite par $V : \text{AP} \rightarrow 2^S$.

Définition 10.14. Soit $M = (S, \text{Act}, \rightarrow, \text{AP}, L)$ un modèle de Kripke. L'*extension ultrafiltre* de M est

$$\mathfrak{Uf}(M) = \underbrace{(\mathfrak{Uf}(S), \text{Act}, \rightarrow, \text{AP}, L')}_{(S, \text{Act}, \rightarrow)^+},$$

où $L' : \mathfrak{Uf}(S) \rightarrow 2^{\text{AP}}$ est générée par

$$\begin{aligned} V' : \text{AP} &\longrightarrow 2^{\mathfrak{Uf}(S)} \\ a &\longmapsto \{\mathcal{F} \mid V(a) \in \mathcal{F}\}. \end{aligned}$$

Remarque 10.6. Dans $\mathfrak{Uf}(M)$ les propriétés suivantes sont équivalentes :

- ▷ $\mathcal{F} \xrightarrow{\alpha} \mathcal{H}$

- ▷ pour tout $A \in \wp(S)$, $A \in \mathcal{H}$ implique $\llbracket \langle \alpha \rangle \rrbracket(A) \in \mathcal{F}$;

- ▷ pour tout $A \in \wp(S)$, $\llbracket \llbracket \alpha \rrbracket \rrbracket(A) \in \mathcal{F}$ implique $A \in \mathcal{H}$.

Remarque 10.7. Si M est *fini* alors

- ▷ $\pi : S \rightarrow \mathfrak{Uf}(S)$ est une bijection ;
- ▷ $\pi(s) \xrightarrow{\alpha} \pi(t)$ ssi $s \xrightarrow{\alpha} t$;
- ▷ $L'(\pi(s)) = L(s)$.

Proposition 10.4. Pour tout $\mathcal{F} \in \mathfrak{Uf}(S)$, et tout ϕ une formule HML, on a

$$\underbrace{\llbracket \phi \rrbracket}_{\text{dans } M} \in \mathcal{F} \iff \underbrace{\mathcal{F} \Vdash \phi}_{\text{dans } \mathfrak{Uf}(M)}.$$

Preuve. On procède par induction sur ϕ .

Cas $a \in \text{AP}$. Par définition de $\mathfrak{Uf}(M)$, on a $\mathcal{F} \Vdash a$ ssi $\llbracket a \rrbracket \in \mathcal{F}$.

Cas \neg . Vu dans le *Homework*, partie II.

Cas $\langle \alpha \rangle$. On montre que $\mathcal{F} \Vdash \langle \alpha \rangle \phi \iff \llbracket \langle \alpha \rangle \rrbracket(\llbracket \phi \rrbracket) \in \mathcal{F}$.

▷ $\llbracket \langle \alpha \rangle \rrbracket(\llbracket \phi \rrbracket) \in \mathcal{F}$. Si $\mathcal{F} \xrightarrow{\alpha} \mathcal{H}$ tel que $\mathcal{H} \Vdash \phi$ alors, par hypothèse d'induction $\llbracket \phi \rrbracket \in \mathcal{H}$ et donc $\llbracket \langle \alpha \rangle \rrbracket(\llbracket \phi \rrbracket) \in \mathcal{F}$.

▷ $\llbracket \langle \alpha \rangle \rrbracket(\llbracket \phi \rrbracket) \in \mathcal{F}$. Supposons $\llbracket \langle \alpha \rangle \rrbracket(\llbracket \phi \rrbracket) \in \mathcal{F}$. On va montrer qu'il existe \mathcal{H} tel que $\mathcal{F} \xrightarrow{\alpha} \mathcal{H}$ et $\llbracket \phi \rrbracket \in \mathcal{H}$. On va utiliser le lemme de l'ultrafiltre. Soit

$$H := \left\{ A \cap \llbracket \phi \rrbracket \mid \begin{array}{l} A \in \wp(S) \\ \text{et} \\ \llbracket \alpha \rrbracket(A) \in \mathcal{F} \end{array} \right\}.$$

1. Remarquons que H est stable par intersections binaires : si $A_1 \cap \llbracket \phi \rrbracket, A_2 \cap \llbracket \phi \rrbracket \in H$ alors $(A_1 \cap A_2) \cap \llbracket \phi \rrbracket \in H$ car $\llbracket \alpha \rrbracket(A_1), \llbracket \alpha \rrbracket(A_2) \in \mathcal{F}$ et donc

$$\llbracket \alpha \rrbracket(A_1 \cap A_2) = \llbracket \alpha \rrbracket(A_1) \cap \llbracket \alpha \rrbracket(A_2) \in \mathcal{F}.$$

2. De plus, on a que $\emptyset \notin H$. En effet, soit $A \cap \llbracket \phi \rrbracket \in H$ alors

$$\llbracket \langle \alpha \rangle \phi \rrbracket \cap \llbracket \alpha \rrbracket(A) \neq \emptyset$$

(intersection de deux éléments de \mathcal{F}), il existe donc $s \in S$ tel que $s \models \langle \alpha \rangle \phi$ et pour tout $s' \xrightarrow{\alpha} s$, on a $s' \in A$. Donc $A \neq \emptyset$ (car il existe $s' \xrightarrow{\alpha} s$).

3. On a $S \cap \llbracket \phi \rrbracket \in H$ car $\llbracket \alpha \rrbracket(S) = S \in \mathcal{F}$.

Donc H a la propriété des intersections finies, et par le lemme de l'ultrafiltre, il existe $\mathcal{H} \in \mathfrak{Uf}(S)$ tel que $\mathcal{H} \supseteq H$. De plus, on a $\mathcal{F} \xrightarrow{\alpha} \mathcal{H}$ car, pour tout $A \in \wp(S)$, $\llbracket \alpha \rrbracket(A) \in \mathcal{F}$ implique $A \cap \llbracket \phi \rrbracket \in H$ qui implique $A \in \mathcal{H}$. On a aussi $\llbracket \phi \rrbracket \in \mathcal{H}$ car $S \cap \llbracket \phi \rrbracket \in \mathcal{H}$. Ainsi, par hypothèse d'induction, on a $\mathcal{H} \Vdash \phi$ et donc $\mathcal{F} \Vdash \langle \alpha \rangle \phi$.

Cas $[\alpha]$. Par dualité.

□

Corollaire 10.1. Soit ϕ une formule HML. Pour tout $s \in S$, on a

$$s \Vdash \phi \iff s \in \llbracket \phi \rrbracket \iff \llbracket \phi \rrbracket \in \pi(s).$$

De plus,

$$M \models \phi \iff \llbracket \phi \rrbracket = S \iff \mathfrak{Uf}(M) \models \phi.$$

Proposition 10.5. L'extension ultrafiltre $\mathfrak{Uf}(M)$ de M est modalement saturée. En particulier, étant donnés M_1 et M_2 deux modèles de Kripke, on a

$$s_1 \equiv s_2 \iff \pi(s_1) \equiv \pi(s_2) \iff \pi(s_1) \sim_{\mathfrak{Uf}(M)} \pi(s_2).$$