Automates temporisés

– introduction par un néophyte –

Partie I / II – Mots et automates temporisés

Mercredi 30 octobre 20002 – ÉNS Lyon

Jérôme DURAND-LOSE

jerome.durand-lose@ens-lyon.fr

MC2

LIP - ÉNS Lyon

Plan

1. Automates (non temporisés)
   (a) Mots / langages / Automates finis
   (b) Extensions non temporisées (mots infinis)
2. Mots et langages temporisés
   (a) Définitions
   (b) Opérations, propriétés
3. Automates temporisés
   (a) Définitions, exemple
   (b) Automate des régions
4. Quelques complexités
   (a) Non clos par complémentaire
   (b) Vacuité est PSPACE-complet
   (c) Universalité est co-$\mathcal{R}.\mathcal{E}.$-complet

Automates Finis (non temporisés)

1. Mots / langages

2. Automates finis

3. Utilités

Mots, langages et leurs opérations

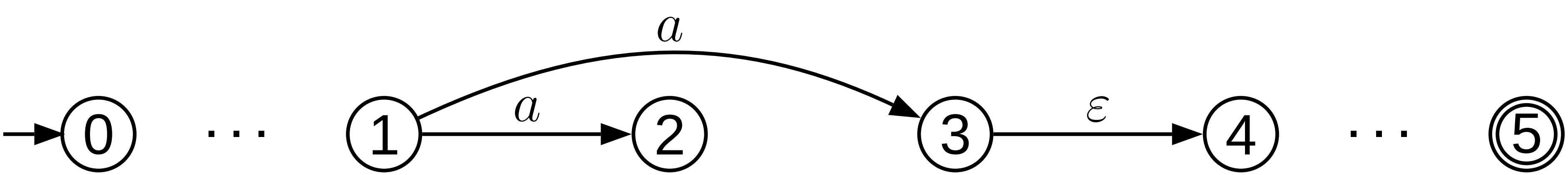
- *Alphabet* : ensemble fini Σ (*e.g.* $\{a, b, c\}$)
- *Mot* : suite finie de lettre (*e.g.* $aabb$)
  Opération : concaténation (*e.g.* $aabb \cdot dc = aabbdc$)
  (monoïde libre)
- *Langage* : ensemble de mots
  Opérations ensemblistes : $\complement L$, $L_1 \cup L_2$, $L_1 \cap L_2$,...
  Opérations particulières : $L_1 \cdot L_2$, L^*,...
- Langages rationnels (*regular*), expression rationnelle...
  Lemme et Théorème de l'étoile

Automates finis

$\mathcal{A} = (\Sigma, Q, I, F, \Delta)$

Q : ensemble fini d'états (I : initiaux, F : acceptants)

Δ : ensemble des transitions



Reconnaissance… $\mathcal{L}(\mathcal{A})$

- Théorème de Kleene
- Constructions pour $\mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$, $\mathcal{L}(\mathcal{A}_1) \cdot \mathcal{L}(\mathcal{A}_2)$, $\mathcal{L}(\mathcal{A})^*$, $\mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$, …
- Déterminisation, minimisation, $\complement\mathcal{L}(\mathcal{A})$,…

Utilités

- Langage
  - Reconnaissance de motif, d'information
  - Compilation, éléments lexicographiques
- Automatisme
  - Modélisation de systèmes
  - Vérification
- Extensions nécessaires
  - Comportement sans fin ?
  - Temporisation ?

ω-langages

- *ω-mot* : suite infinie de lettres
  ω-langage : ensemble d'ω-mots
- *ω-automate* (seule la longueur du parcours change)

  Inf : états infiniment visités par un parcours

	Büchi	Muller
	$F \subseteq Q$	$\mathcal{F} \subseteq \mathcal{P}(Q)$
Accepté *ssi*	*Inf* $\cap F \neq \emptyset$	*Inf* $\in \mathcal{F}$

  Büchi déterministe $\subsetneq$ Büchi = Muller (déter. ou non)

- *ω-expression régulière* : ($<$ e.r.$>$) $\cdot$ ($<$ e.r.$>$)$^{\omega}$

  (Mc.Nauton) équivalent Muller

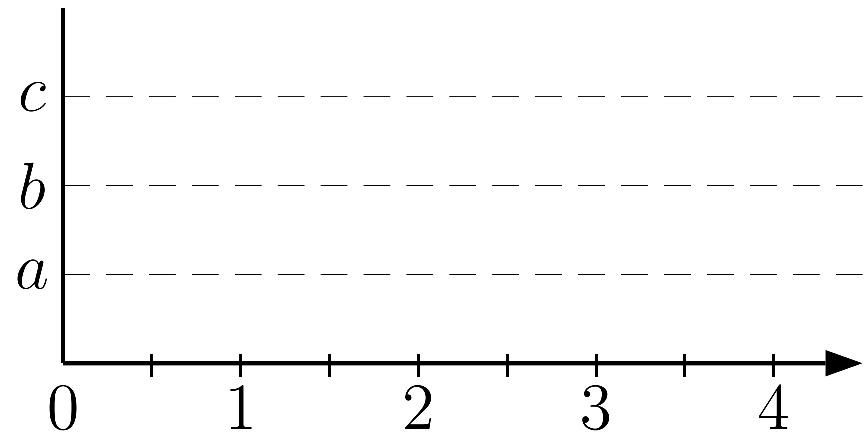
Langages sur les ordinaux

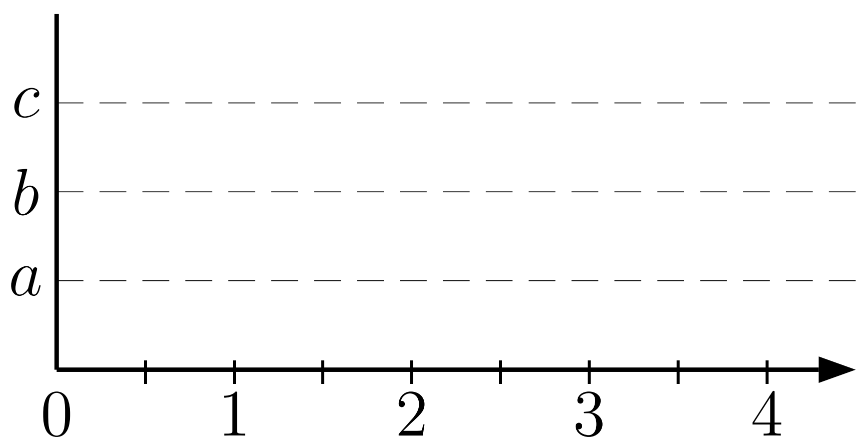
- Mot / ordinaux :
  suite infinie de lettres indexée par des ordinaux
  $0,\ 1, \ldots \omega,\ \omega{+}1, \ldots\ 2\omega,\ 2\omega{+}1, \ldots\ 3\omega \ldots$
  $\omega^2,\ \omega^2{+}1, \ldots \omega^2{+}\omega,\ \omega^2{+}\omega{+}1, \ldots\ \omega^2{+}2\omega,\ \omega^2{+}2\omega{+}1, \ldots$
  $\omega^3, \ldots$
  $\ldots 7\omega^6 + 5\omega^3 \ldots$
- Définition de langages, d'automates, d'é.r.. . .
- Pas mal étudiés au siècle dernier :
  Büchi, Choueka, Hermmer & Volper, Kleene,
  Wojciechowski

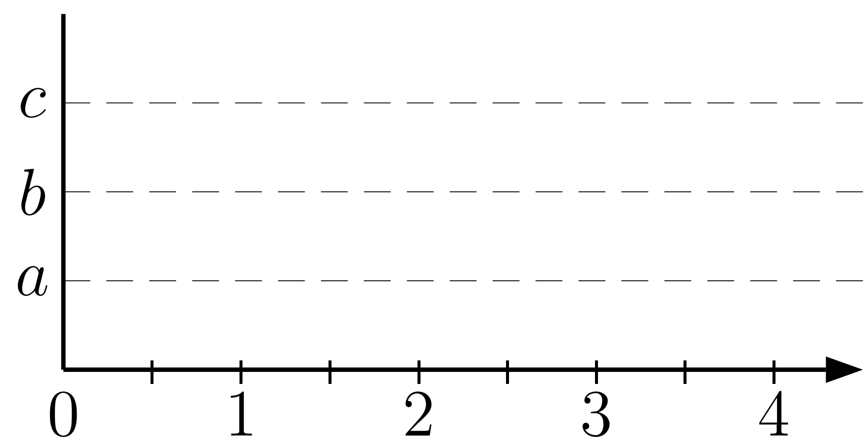
Mots temporisés

Mot non temporisé :

$abac$





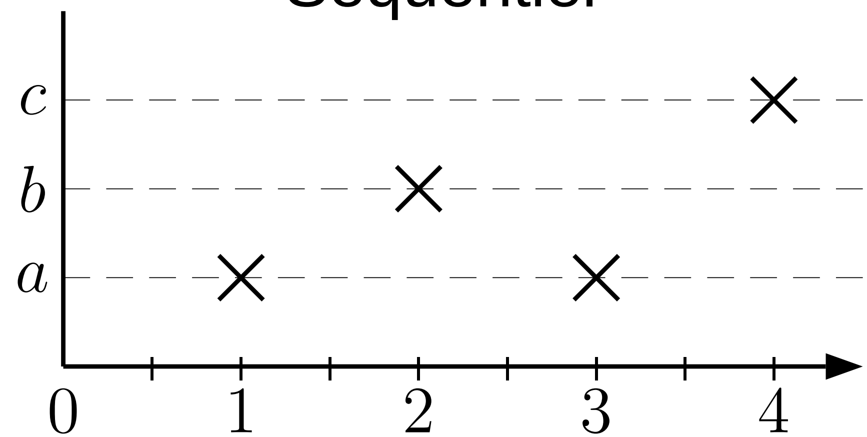


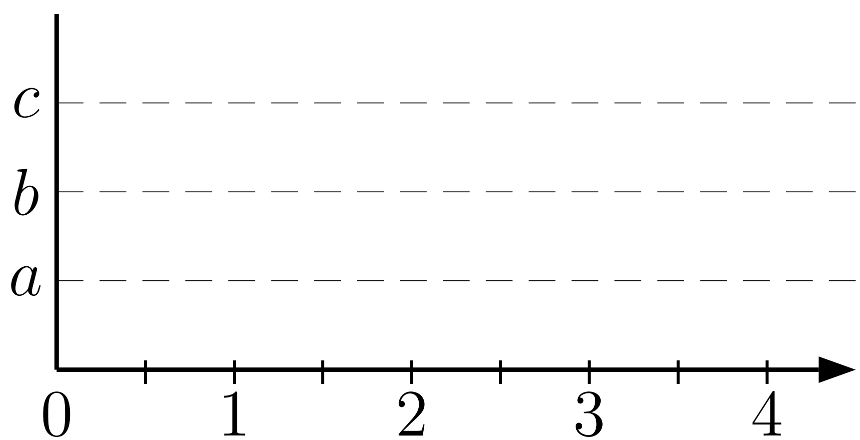
Mots temporisés

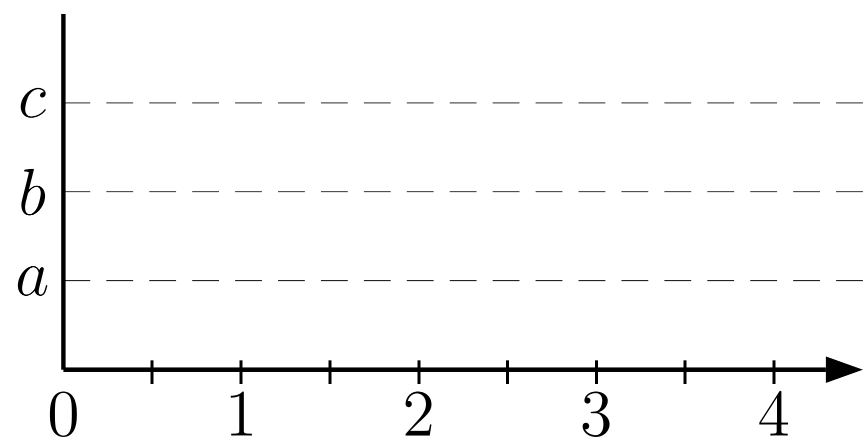
Mot non temporisé :

$abac$

Séquentiel





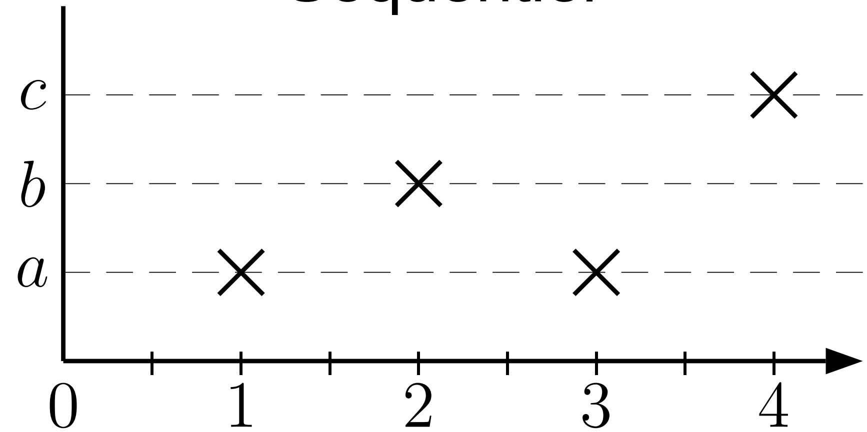


Mots temporisés

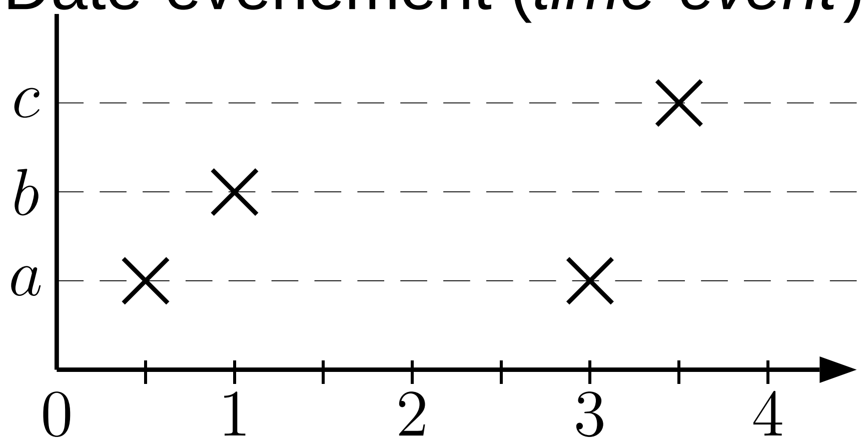
Mot non temporisé :

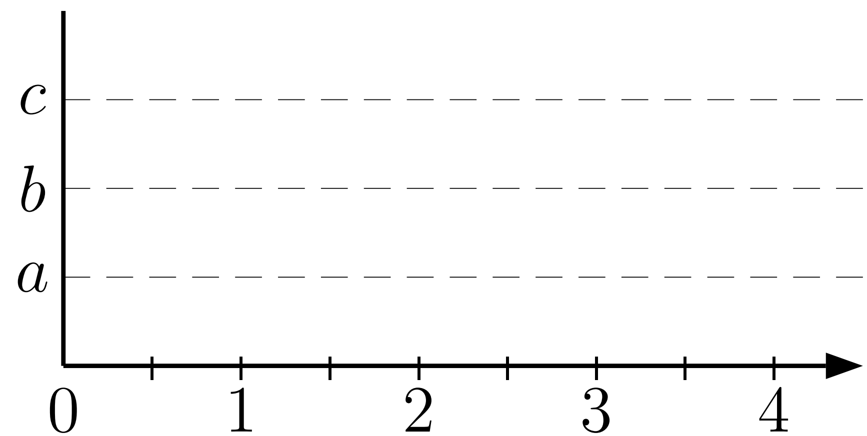
$abac$

Séquentiel



Date-évènement (*time-event*)





$$(a, 0.5)(b, 1)(a, 3)(c, 3.5)$$

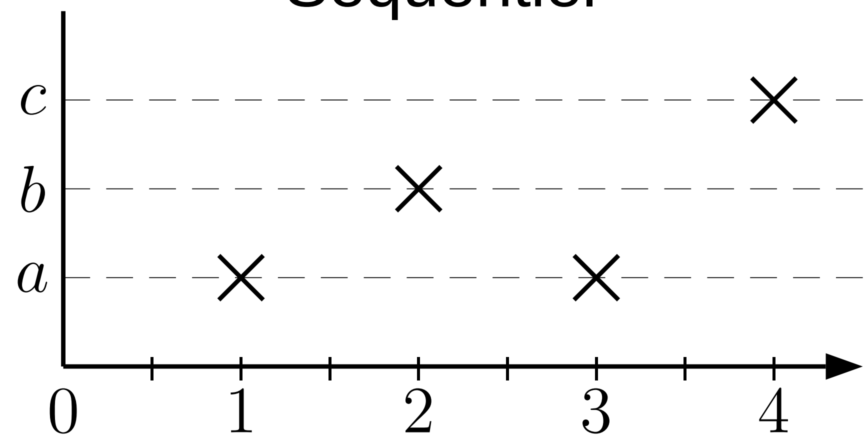
$$a^{0.5}b^{0.5}a^{2}c^{0.5}$$

Mots temporisés

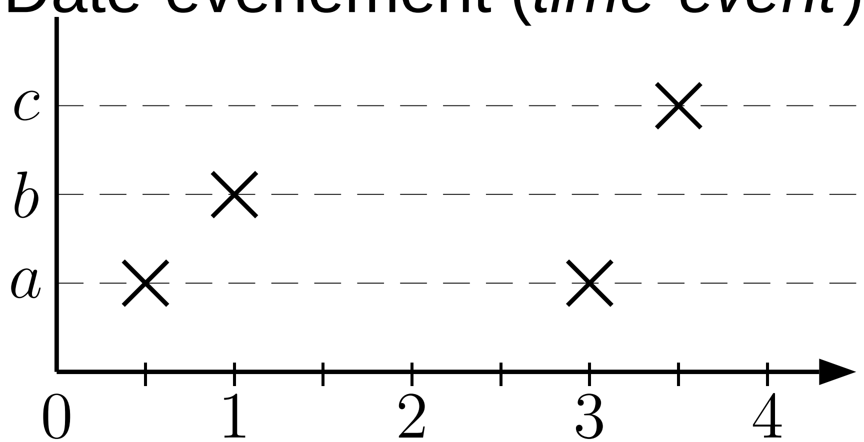
Mot non temporisé :

$abac$

Séquentiel



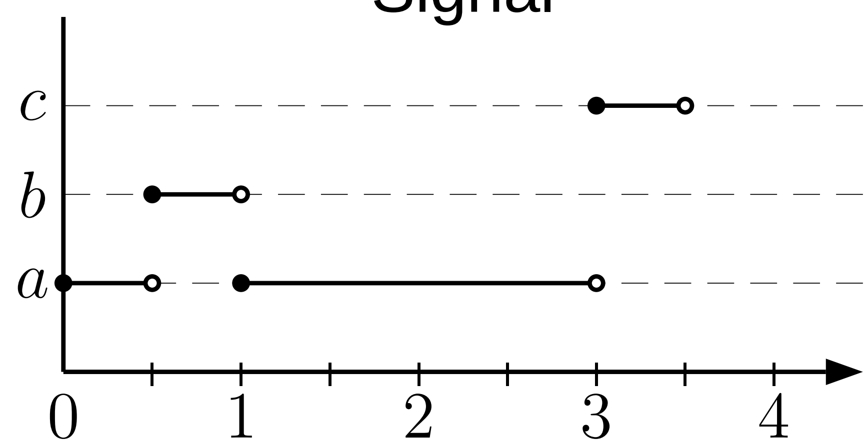
Date-évènement (*time-event*)



$(a, 0.5)(b, 1)(a, 3)(c, 3.5)$

$a^{0.5}b^{0.5}a^2c^{0.5}$

Signal



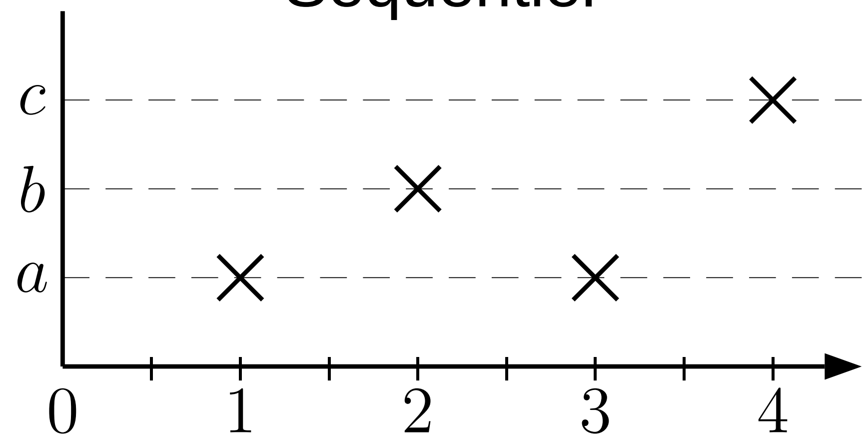
$a^{0.5}b^{0.5}a^2c^{0.5}$

Mots temporisés

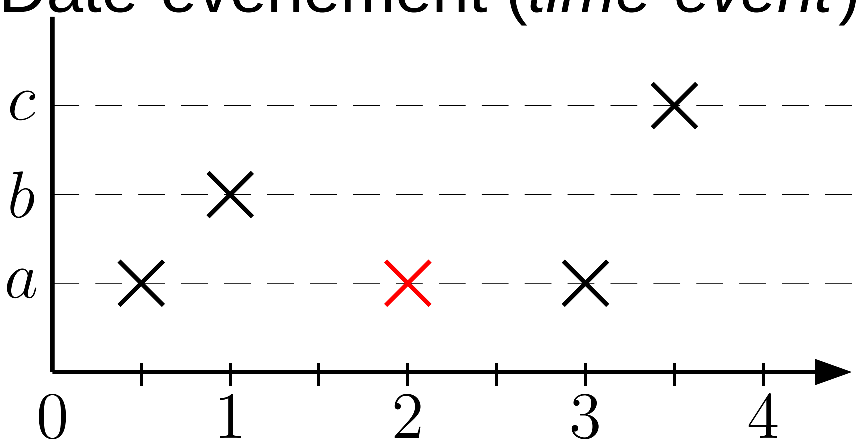
Mot non temporisé :

$abac$

Séquentiel



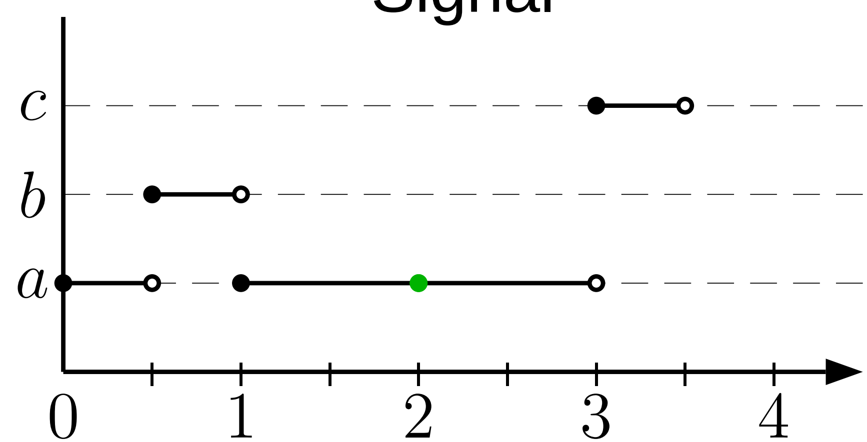
Date-évènement (*time-event*)



$(a, 0.5)(b, 1)(a, 3)(c, 3.5)$

$a^{0.5}b^{0.5}a^{2}c^{0.5} \neq a^{0.5}b^{0.5}a^{1}a^{1}c^{0.5}$

Signal



$a^{0.5}b^{0.5}a^{2}c^{0.5}$

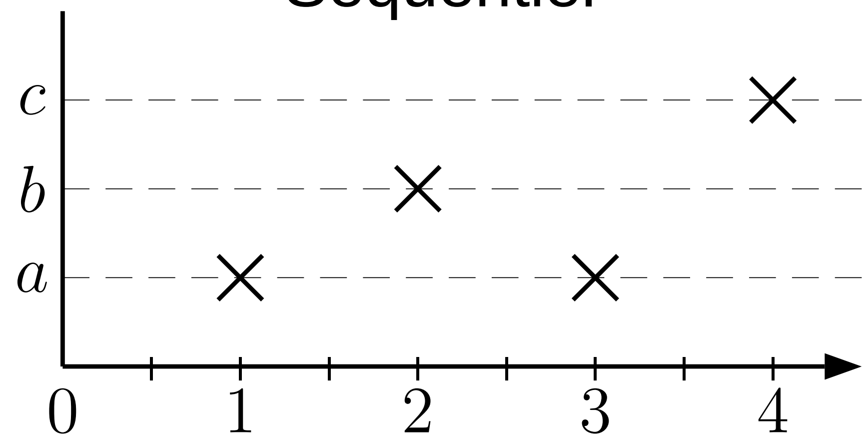
$= a^{0.5}b^{0.5}a^{1}a^{1}c^{0.5}$

Mots temporisés

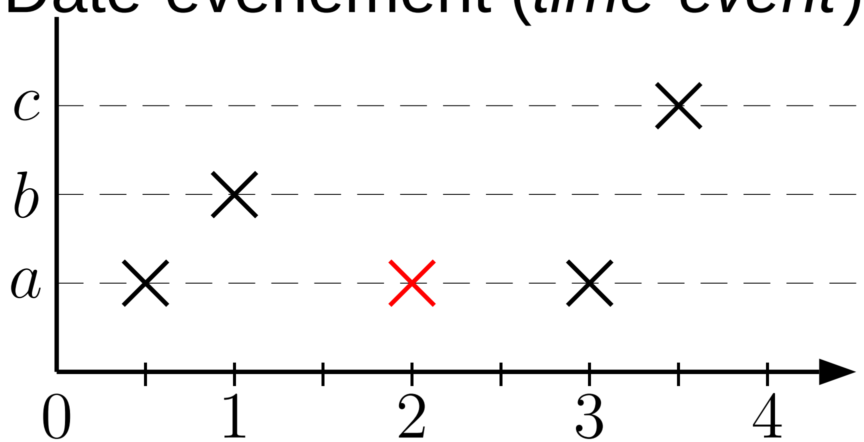
Mot non temporisé :

$abac$

Séquentiel



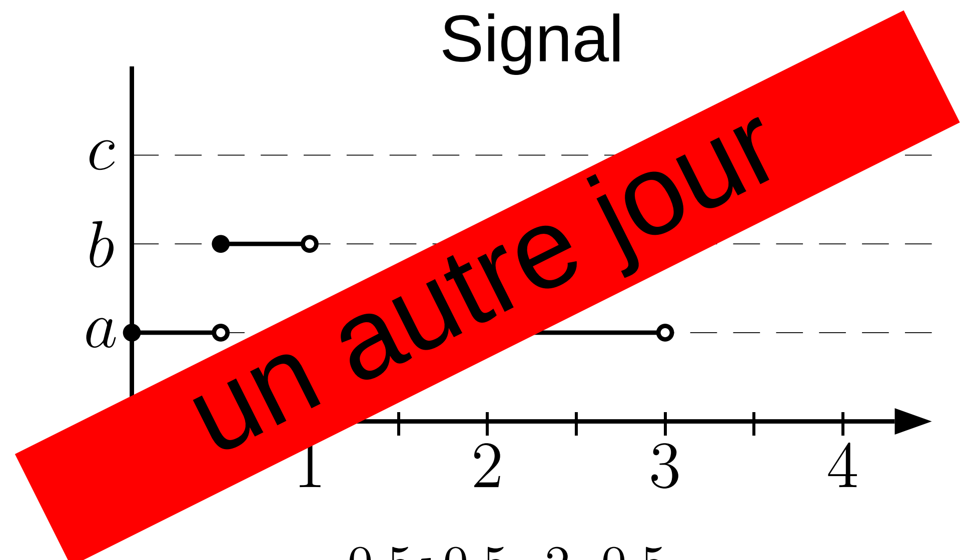
Date-évènement (*time-event*)



$(a, 0.5)(b, 1)(a, 3)(c, 3.5)$

$a^{0.5}b^{0.5}a^{2}c^{0.5} \neq a^{0.5}b^{0.5}a^{1}a^{1}c^{0.5}$

Signal



$a^{0.5}b^{0.5}a^{2}c^{0.5}$

$= a^{0.5}b^{0.5}a^{1}a^{1}c^{0.5}$

Langages temporisés et opérations

- *Durée* d'un mot
  $|(a,0.5)(b,1)(a,3)(c,3.5)| = 3.5$
- *Concaténation classique*
  $(a,0.5)(b,1)(a,3)(c,3.5) \bullet (a,0.5)(b,1)(a,3)(c,3.5)$
  $= (a,0.5)(b,1)(a,3)(c,3.5)(a,4)(b,4.5)(a,6.5)(c,7)$
- *Concaténation superposition* si consécutive / disjointe
  $(a,0.5)(b,1)(a,3)(c,3.5) \circ (a,0.5)(b,1)(a,3)(c,3.5)$ indéfini
  $(a,0.5)(b,1)(a,3)(c,3.5) \circ (a,4)(b,4.5)(a,6.5)(c,7)$
  $= (a,0.5)(b,1)(a,3)(c,3.5)(a,4)(b,4.5)(a,6.5)(c,7)$
- *Langage temporisé* : ensemble de mots temporisés
  Opérations
  - $\bullet$, $\circ$ et deux étoiles de KLEENE : $*$ et $\circledast$
  - Filtre sur durée : $\langle L \rangle_I = \{m \in l \mid |m| \in I\}$
    $\rightsquigarrow$ Aspects algébriques [Asarin et al., 2002]

Discret $\leftrightsquigarrow$ continu

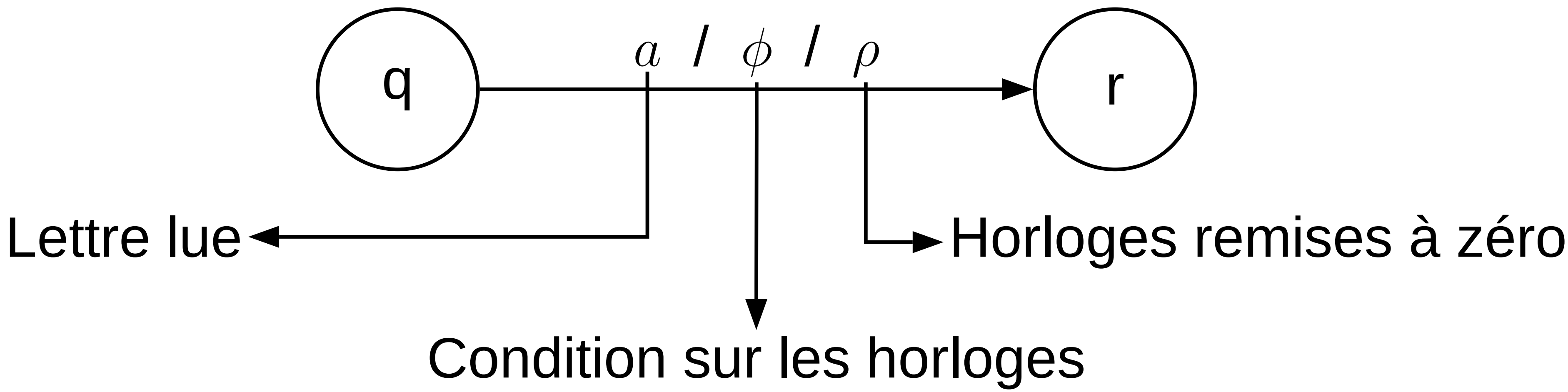
- Successions simultanées
  $(a, 0.5)(a, 1)(b, 1)(a, 1)(a, 2) \ldots$
  on peut l'exclure ou non
- Mots infinis pour comportement sans fin
  $\rightsquigarrow$ ω-mots temporisés
- Possibilité d'accumulation(s)
  $(a, 0.9)(a, .99)(b, .999)(a, .9999)(a, .99999) \ldots$
- Interdiction des *configurations Zénon*
  sinon $\rightsquigarrow$ mots temporisés sur des ordinaux
  [Bérard and Picaronny, 2000]

Automate temporisé

$\mathcal{A} = (\Sigma, Q, I, F, Z, \Delta)$

Z : ensemble fini d'*horloges*,

Δ : ensemble fini de transitions

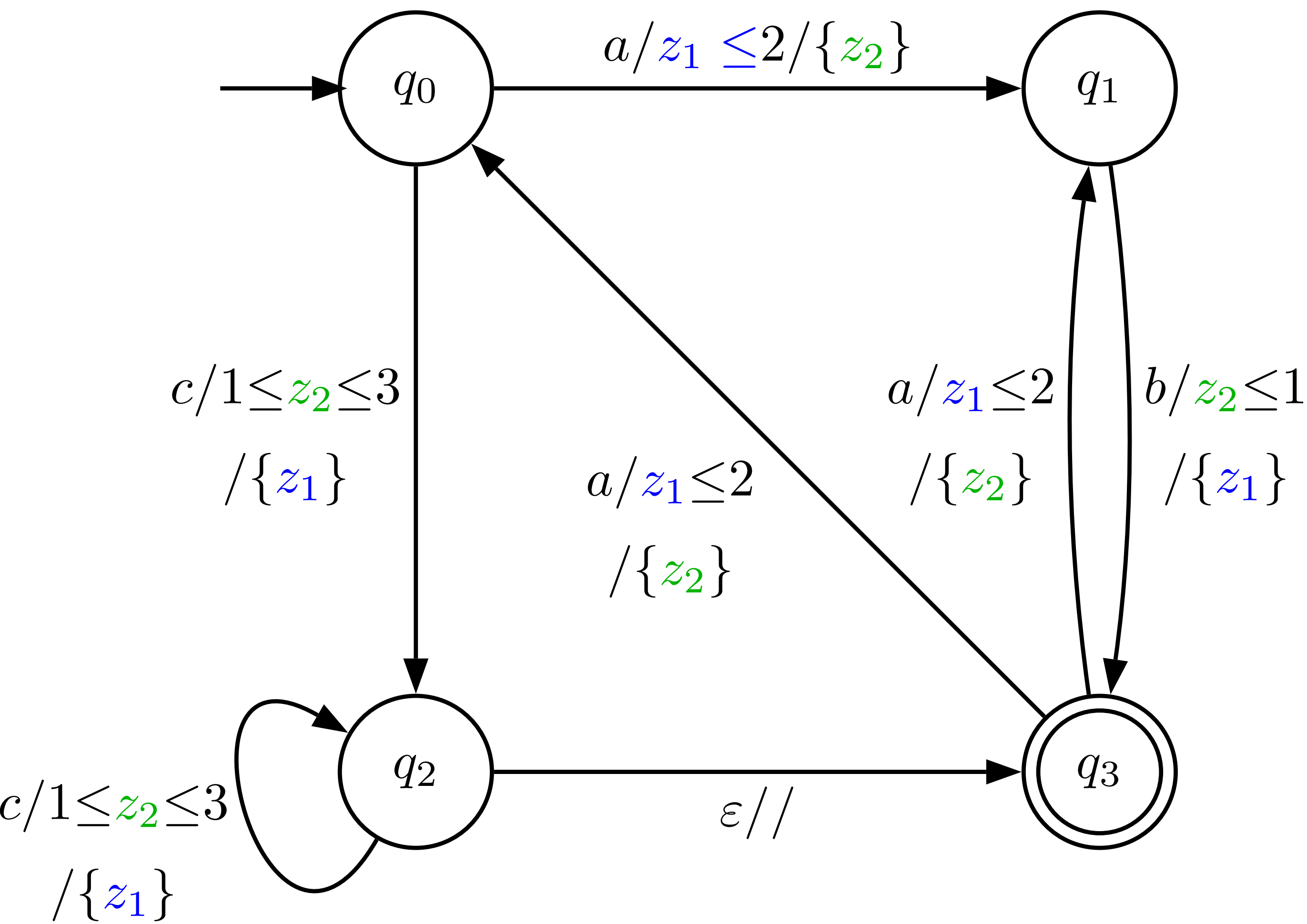


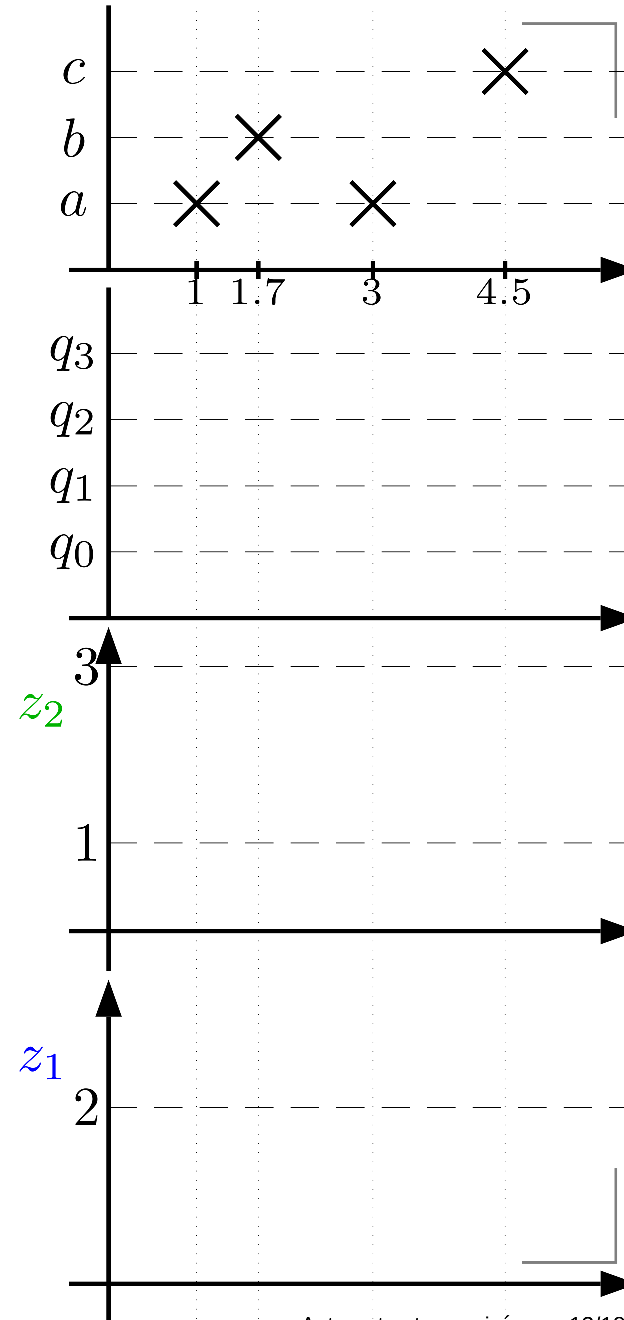
$z \leq c$, $z < c$, $\phi_1 \vee \phi_2$, $\neg\phi$

$c \in \mathbb{R}, \mathbb{Q}, \mathbb{N}$ (clauses plus complexes)

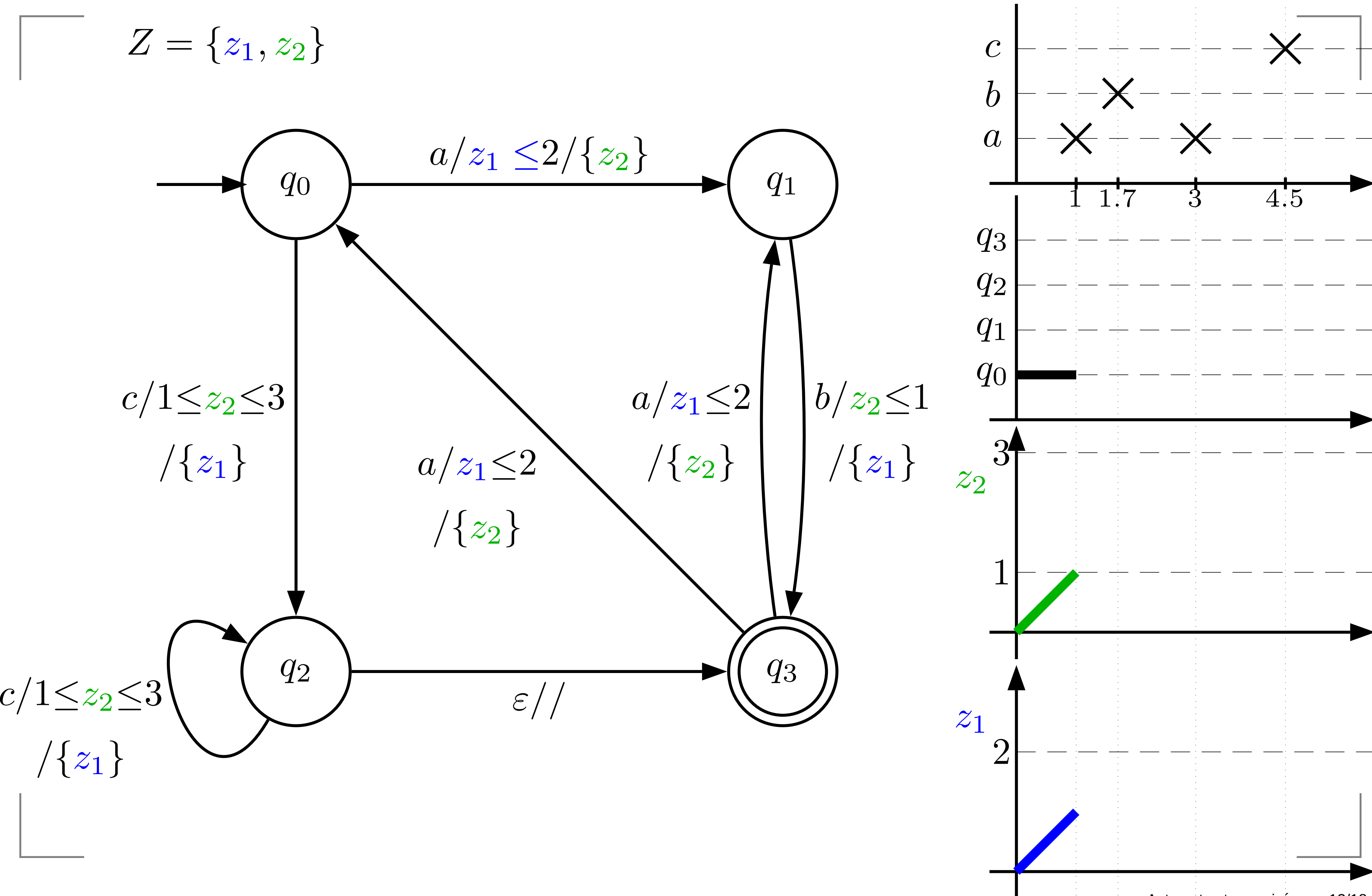
Exemple

$Z = \{z_1, z_2\}$





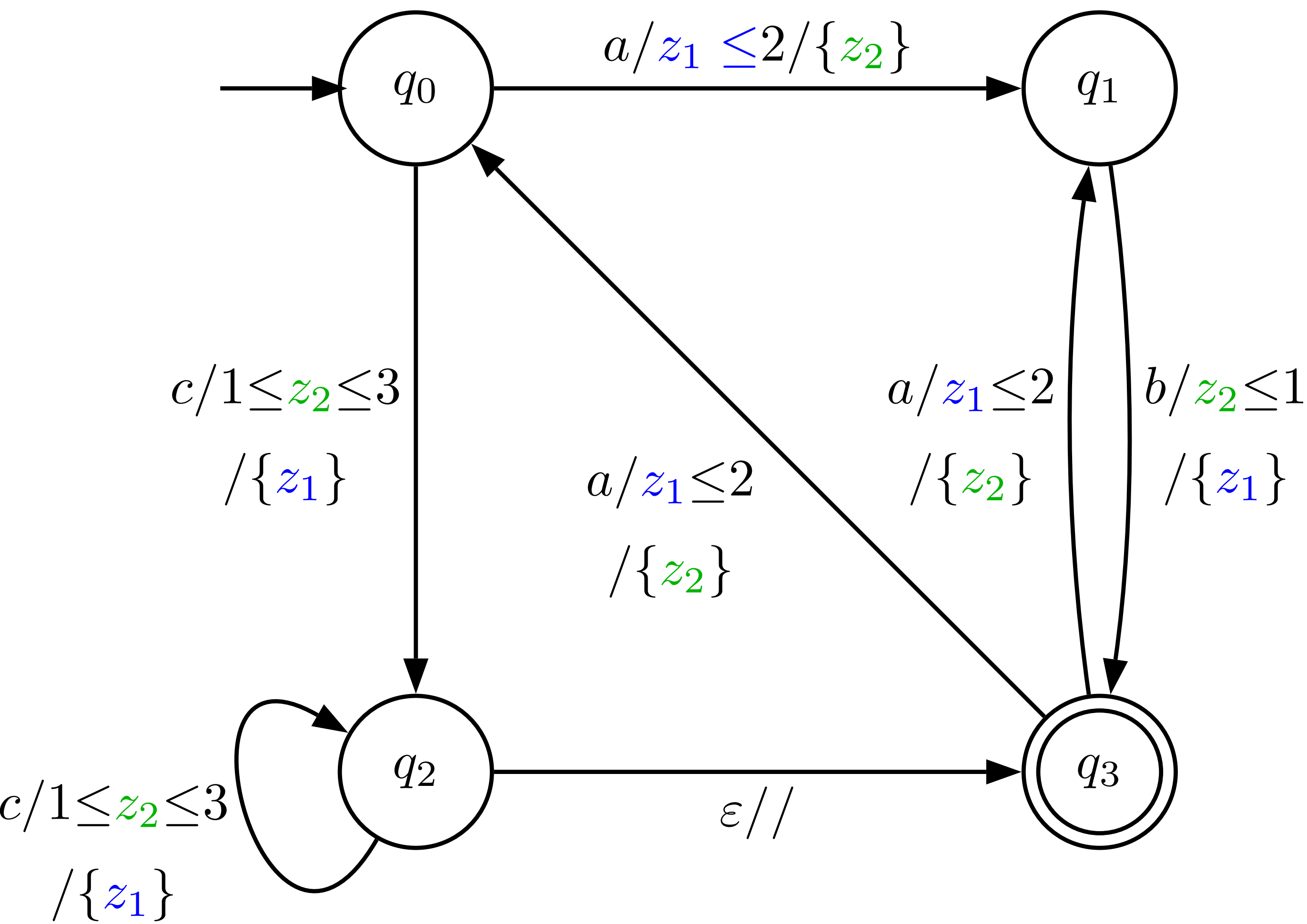
Exemple

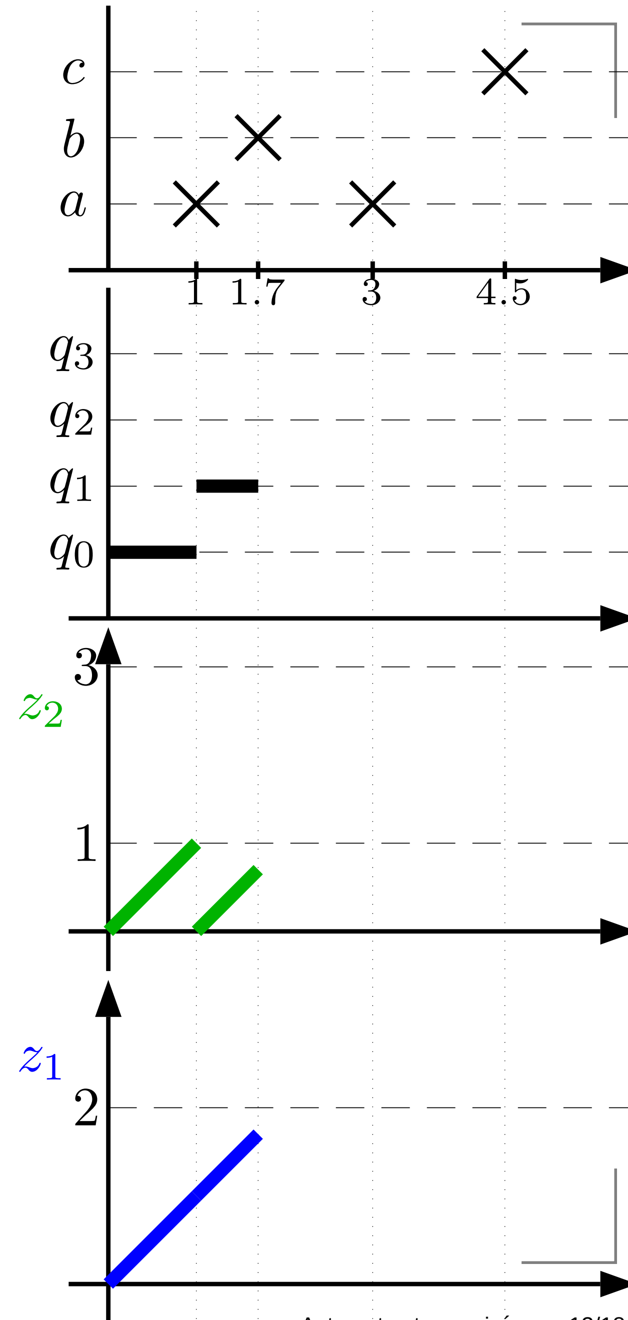


$Z = \{z_1, z_2\}$
$a/z_1 \leq 2/\{z_2\}$
q_0
q_1
q_2
q_3
$c/1\leq z_2\leq 3$
$/\{z_1\}$
$a/z_1\leq 2$
$/\{z_2\}$
$a/z_1\leq 2$
$/\{z_2\}$
$b/z_2\leq 1$
$/\{z_1\}$
$c/1\leq z_2\leq 3$
$/\{z_1\}$
$\varepsilon//$
c
b
a
1
1.7
3
4.5
z_2
3
1
z_1
2

Exemple

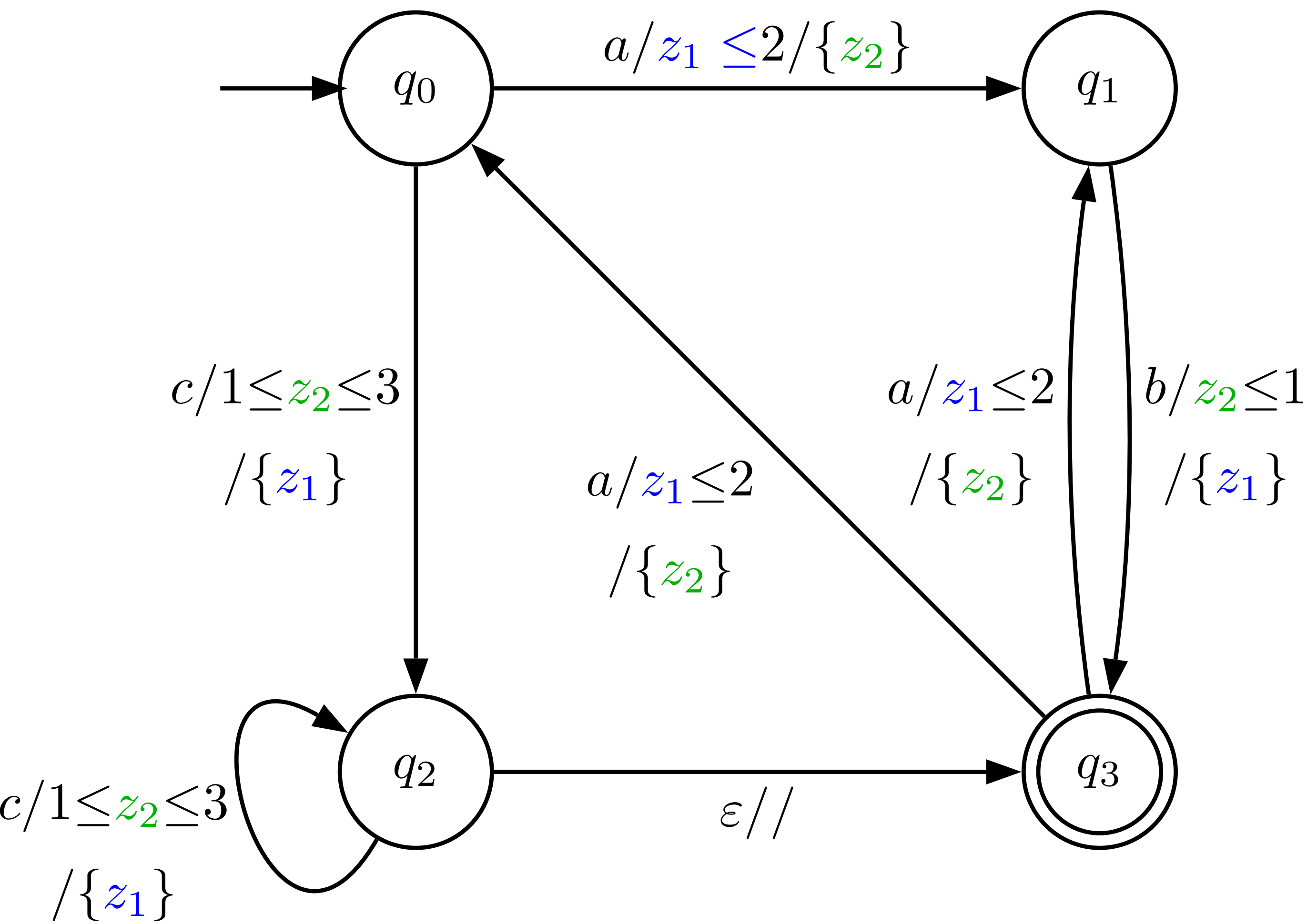
$Z = \{z_1, z_2\}$

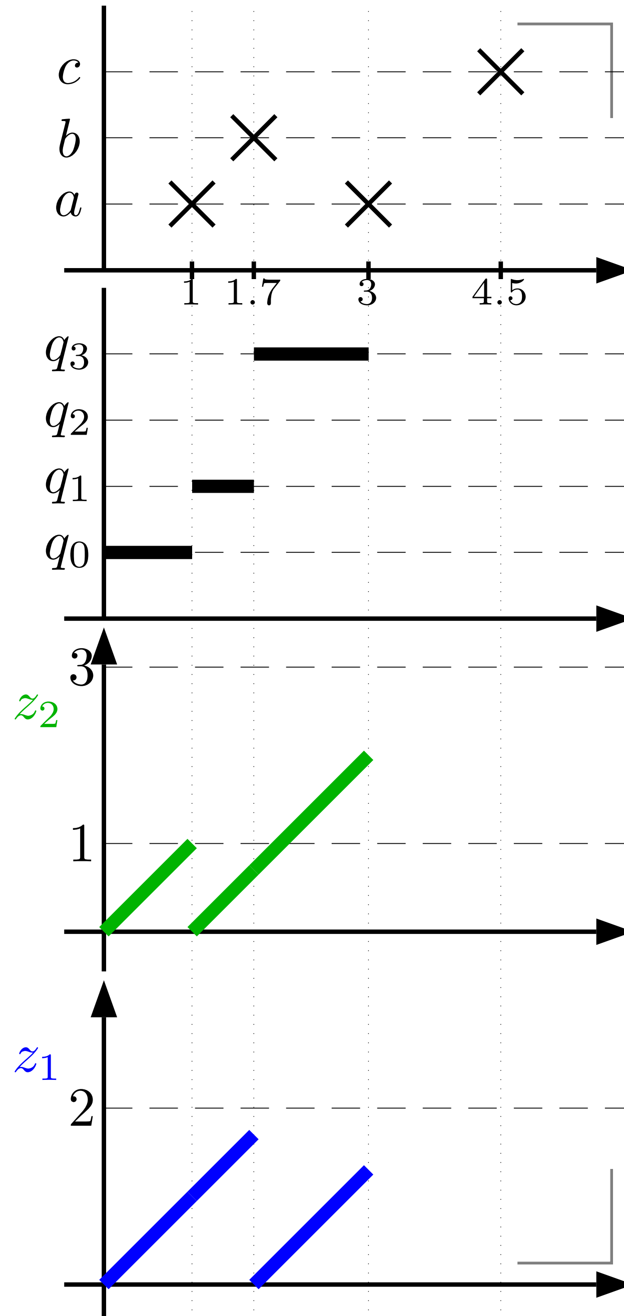




Exemple

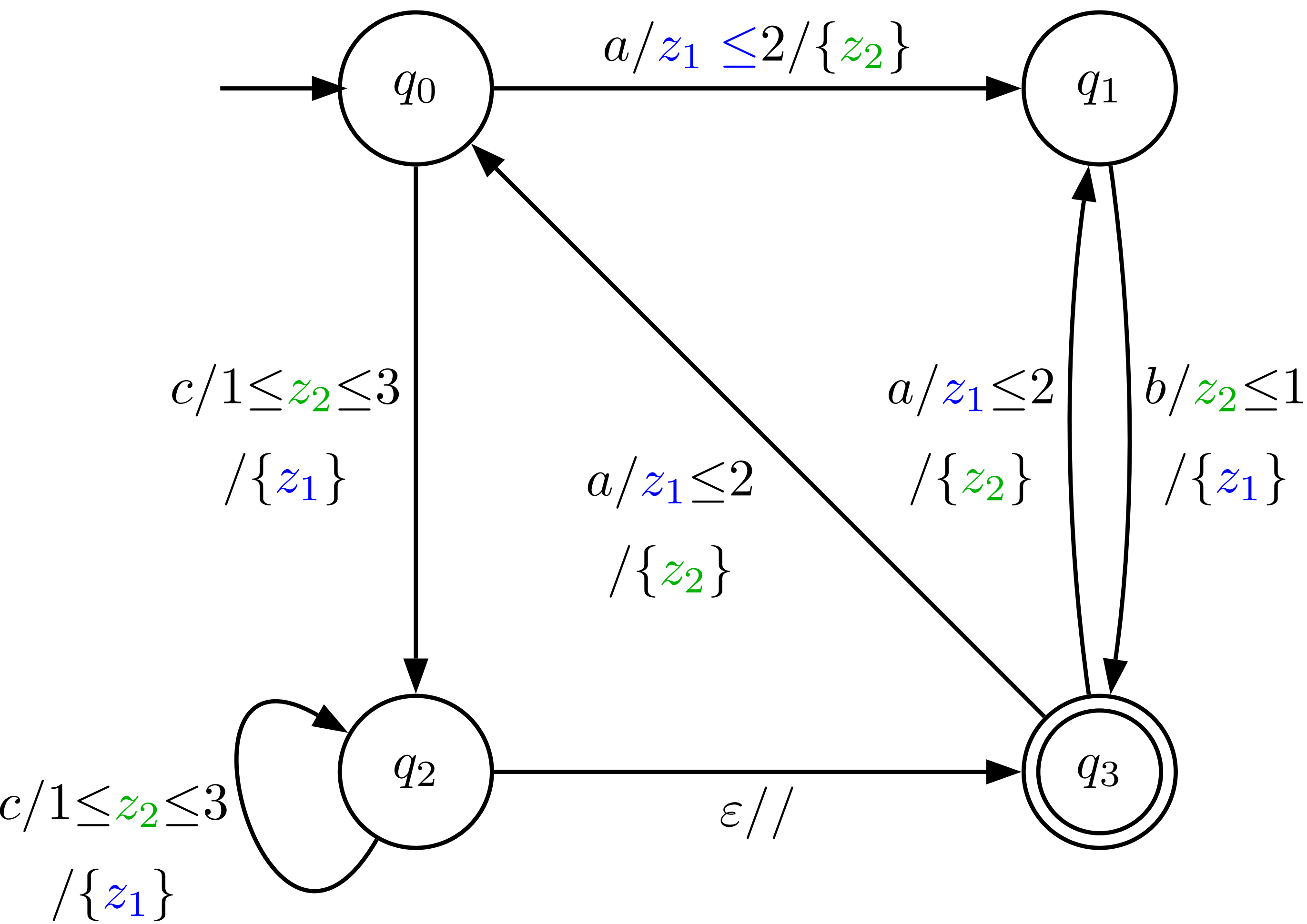
$Z = \{z_1, z_2\}$

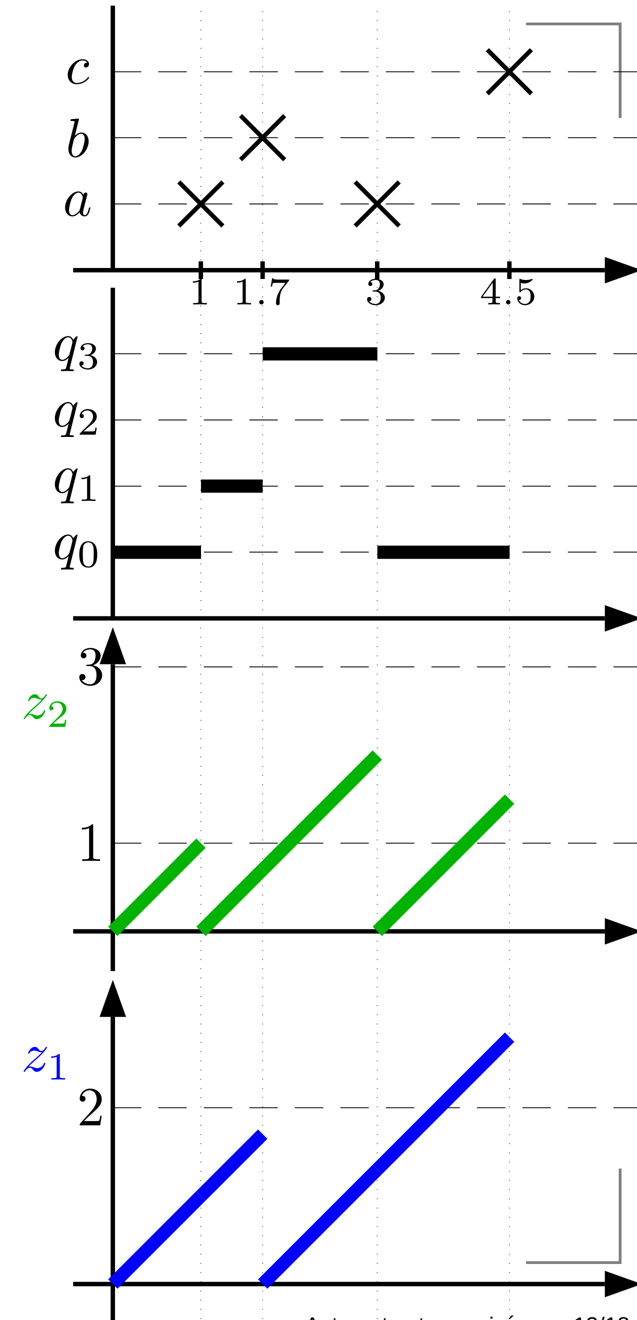




Exemple

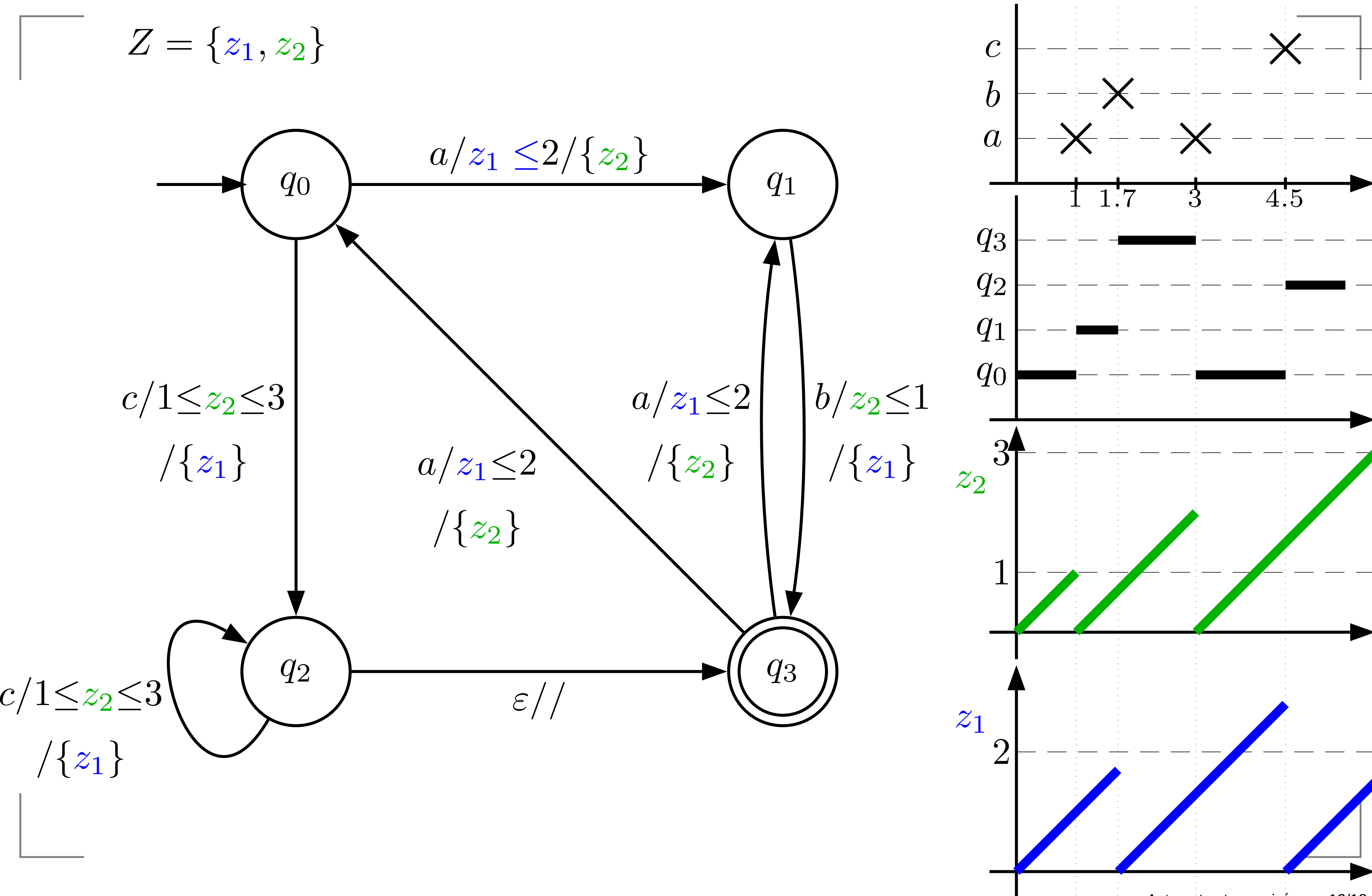
$Z = \{z_1, z_2\}$





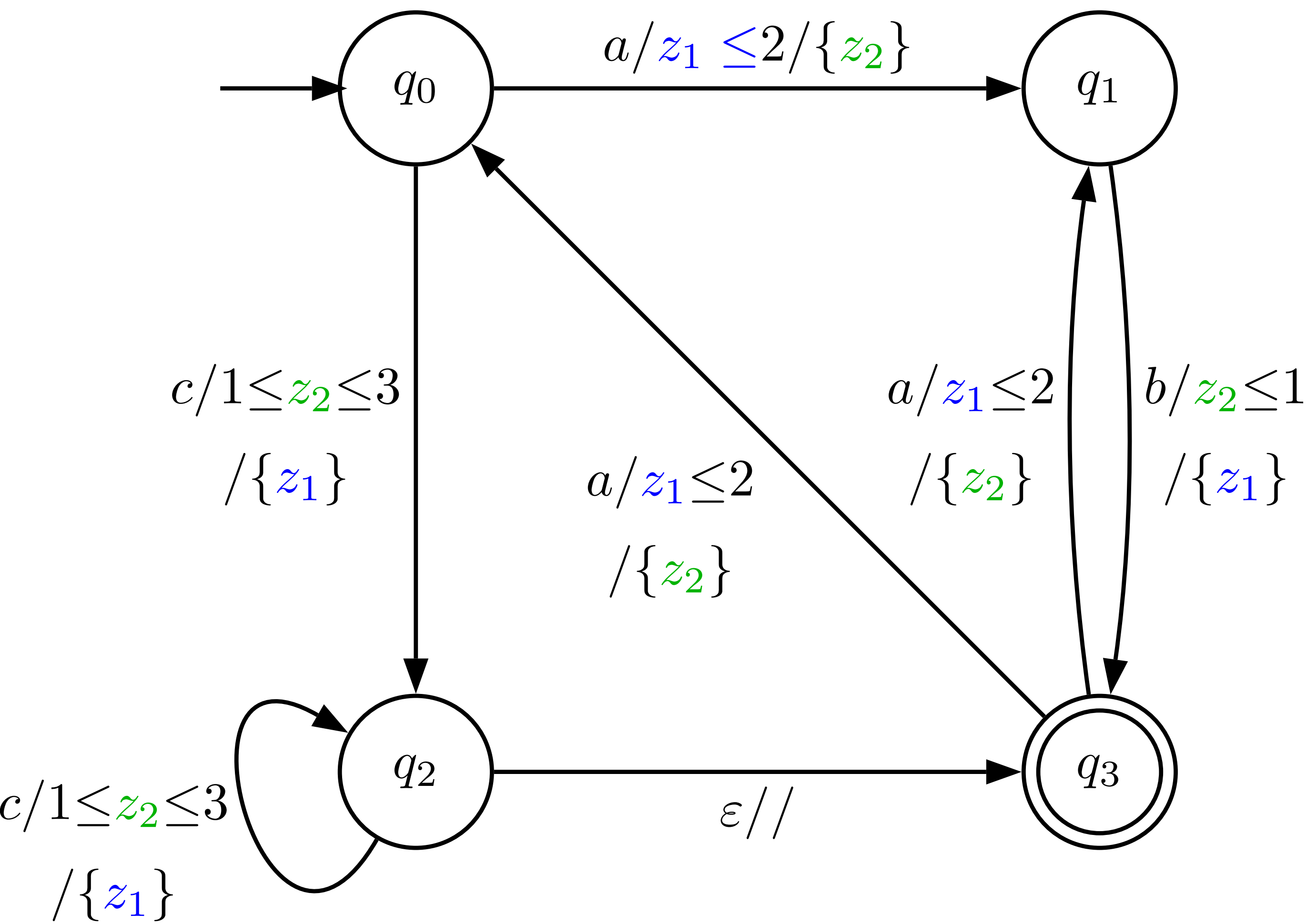
Exemple

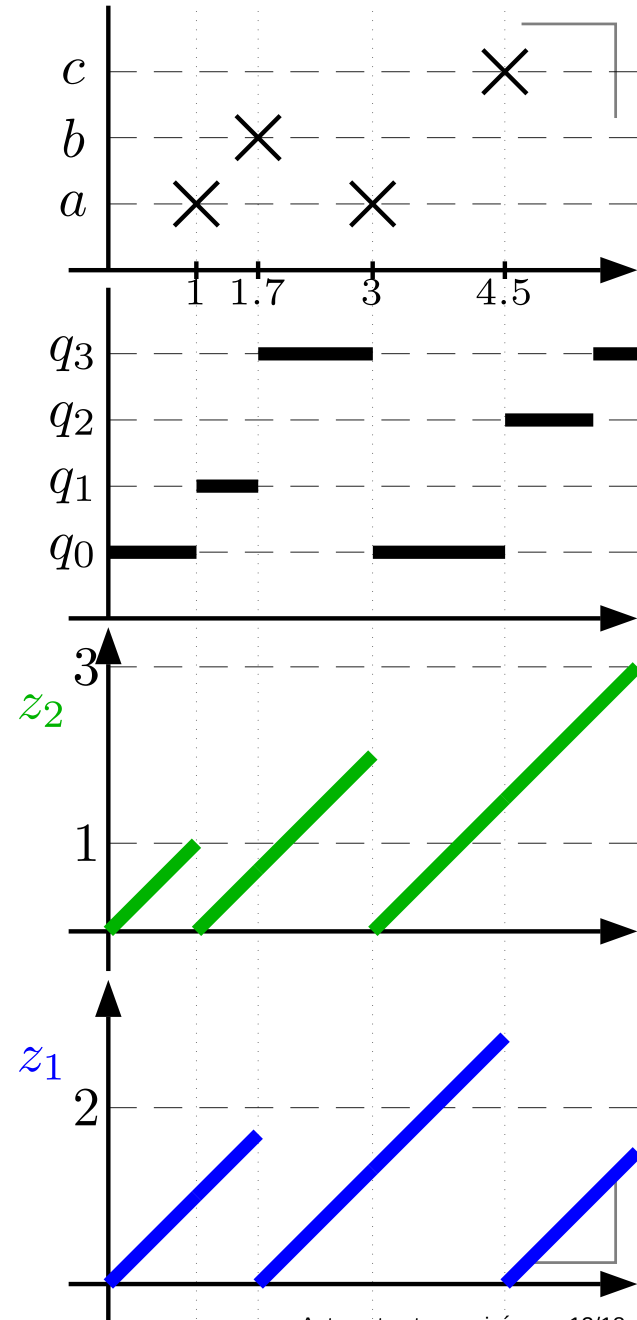
$Z = \{z_1, z_2\}$



Exemple

$Z = \{z_1, z_2\}$





Langage temporisé correspondant ?

- En oubliant toute la partie temporisation :

$$(\,(ab)^+|c^+\,)\,(\,a(\,(ab)^+|c^+\,)\,)^*$$

- Existe-t-il une temporisation pour chacun de ses mots ?

Langage temporisé correspondant ?

- En oubliant toute la partie temporisation :

$$((ab)^+|c^+) (a((ab)^+|c^+))^*$$

- Existe-t-il une temporisation pour chacun de ses mots ?

  Oui, avec des conditions du type :

tout a est au plus à 2 unités de temps après le dernier b ou c précéder
tout b est au plus à 1 unité de temps après le dernier a précédent
tout c est entre 1 et 3 unités de temps après le dernier a précédent

- Liens avec la logique temporelle

Langage temporisé correspondant ?

- En oubliant toute la partie temporisation :

$$(\,(ab)^+|c^+\,)\,(\,a(\,(ab)^+|c^+\,)\,)^*$$

- Existe-t-il une temporisation pour chacun de ses mots ?

  Oui, avec des conditions du type :

tout a est au plus à 2 unités de temps après le dernier b ou c précéden
tout b est au plus à 1 unité de temps après le dernier a précédent
tout c est entre 1 et 3 unités de temps après le dernier a précédent

- Liens avec la logique temporelle

- $<<$Expressions rationnelles$>>$
  [Asarin et al., 2002] [Bouyer and Petit, 2002]

Clos par...

- Union (facile car non déterministe)
- Intersection (produit habituel d'automate)
- Concaténation (recoller en remettant les horloges à 0)
- Superposition consécutive (recoller sans remise à 0)
- Itérations finies ($*$ et $\circledast$)
- Restriction sur la durée (une horloge en plus)

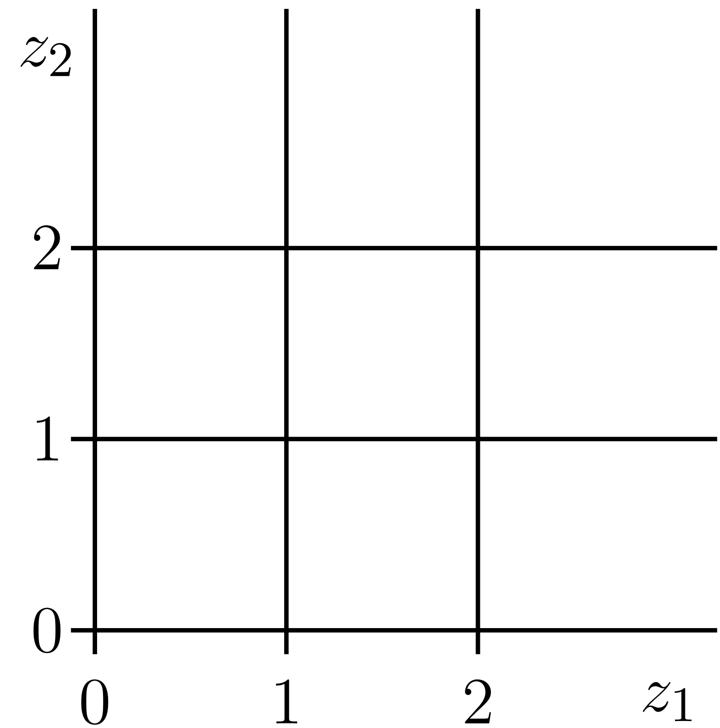
- Complément (?)

- Déterminisation (?)

- Automate minimal (?)

Se débarrasser du temps

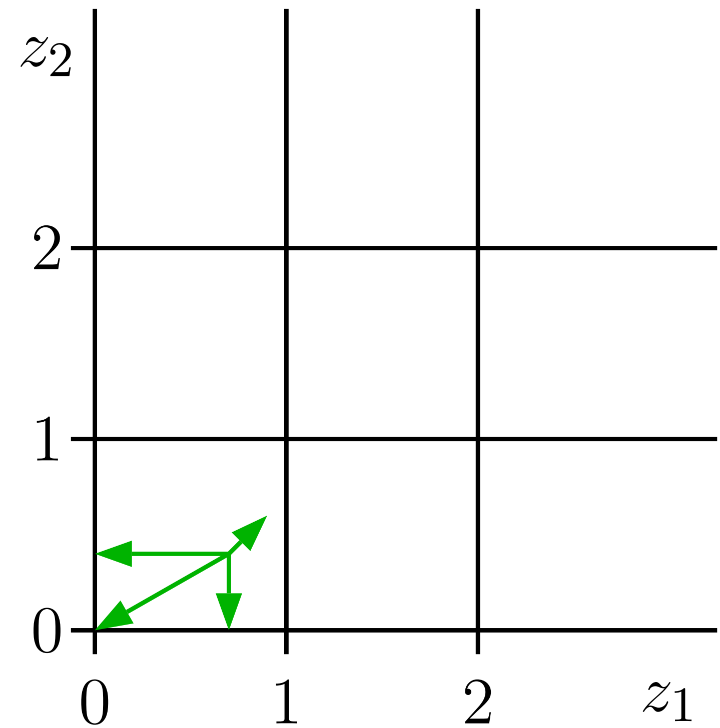
- On suppose les constantes des contraintes dans $\mathbb{Q}$
  En changeant d'échelle elles sont dans $\mathbb{N}$ (dans $[[1, C]]$)
- Découpage de l'espace des temps en *Clock regions*
  Lieux où les contraintes sont constantes
  - Être sur un entier ou entre 2
    $\{0\}$, $(0, 1)$, $\{1\}$, $(1, 2)$, $\{2\}$, $(2, \infty)$



- Augmentation exponentielle de la taille des données
  nombre de séparations de l'ordre de la plus grande constante... écrite en binaire

Se débarrasser du temps

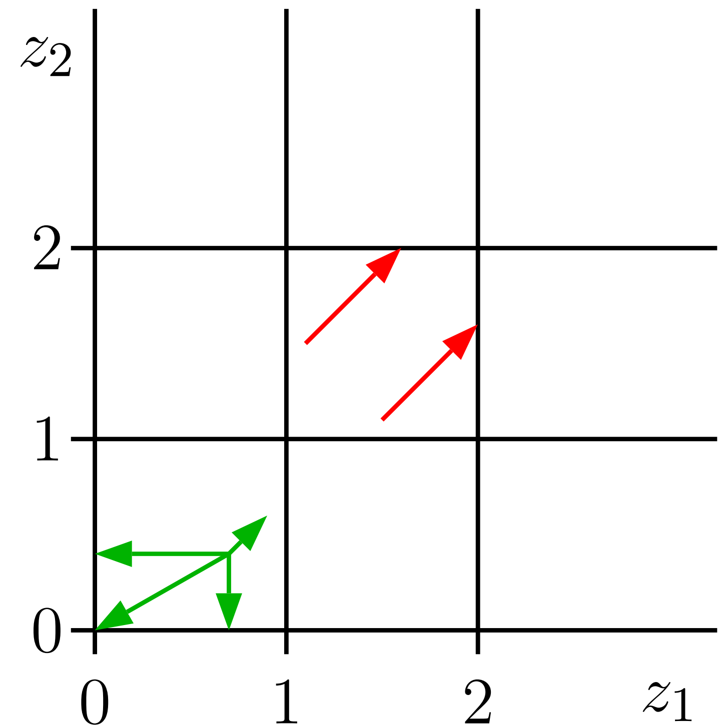
- On suppose les constantes des contraintes dans $\mathbb{Q}$
  En changeant d'échelle elles sont dans $\mathbb{N}$ (dans $[[1, C]]$)
- Découpage de l'espace des temps en *Clock regions*
  Lieux où les contraintes sont constantes
  - Être sur un entier ou entre 2
    $\{0\}$, $(0, 1)$, $\{1\}$, $(1, 2)$, $\{2\}$, $(2, \infty)$
  - Mouvements autorisés :
    - projection(s) (remise(s) à 0)
    - le temps avance suivant $\overline{1}$



Augmentation exponentielle de la taille des données
nombre de séparations de l'ordre de la plus grande constante... écrite en binaire

Se débarrasser du temps

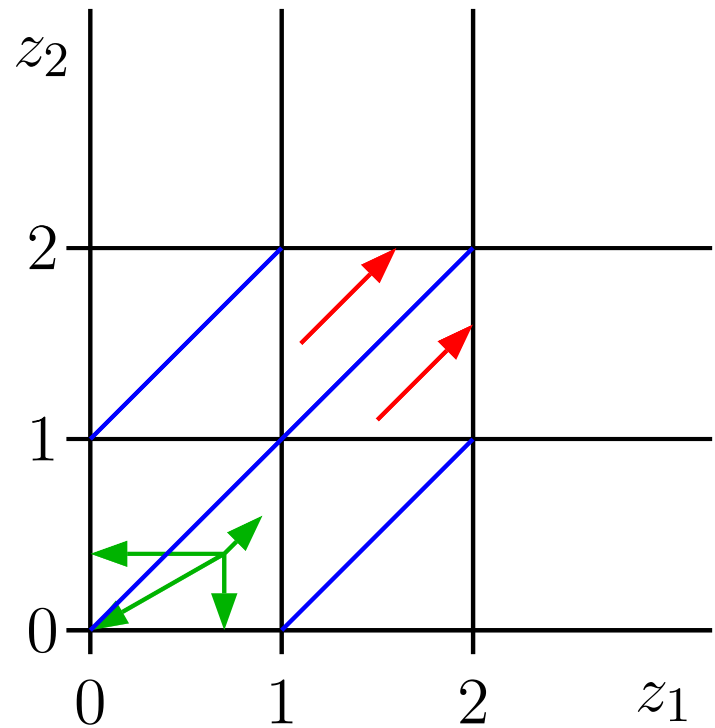
- On suppose les constantes des contraintes dans $\mathbb{Q}$
  En changeant d'échelle elles sont dans $\mathbb{N}$ (dans $[[1, C]]$)
- Découpage de l'espace des temps en *Clock regions*
  Lieux où les contraintes sont constantes
  - Être sur un entier ou entre 2
    $\{0\}$, $(0, 1)$, $\{1\}$, $(1, 2)$, $\{2\}$, $(2, \infty)$
  - Mouvements autorisés :
    - projection(s) (remise(s) à 0)
    - le temps avance suivant $\overline{1}$
  - Problème suivant $\overline{1}$!



- Augmentation exponentielle de la taille des données
  nombre de séparations de l'ordre de la plus grande constante... écrite en binaire

Se débarrasser du temps

- On suppose les constantes des contraintes dans $\mathbb{Q}$
  En changeant d'échelle elles sont dans $\mathbb{N}$ (dans $[[1, C]]$)
- Découpage de l'espace des temps en *Clock regions*
  Lieux où les contraintes sont constantes
  - Être sur un entier ou entre 2
    $\{0\}$, $(0, 1)$, $\{1\}$, $(1, 2)$, $\{2\}$, $(2, \infty)$
  - Mouvements autorisés :
    - projection(s) (remise(s) à 0)
    - le temps avance suivant $\overline{1}$
  - Problème suivant $\overline{1}$!
  - Pré-ordre total sur
    les parties fractionnaires
    $Frac(z_1) \leq = \geq Frac(z_2)$



- Augmentation exponentielle de la taille des données
  nombre de séparations de l'ordre de la plus grande constante... écrite en binaire

Region automaton

- Soit $\mathcal{R}$, l'ensemble des régions
- Les régions α et β se *succèdent*

$$\alpha \preccurlyeq \beta \textit{ ssi } \alpha + \lambda\bar{1} \subseteq \beta \text{ avec } \lambda \text{ positif}$$

- $Q' = Q' \times \mathcal{R}$
  $I' = I = \times\{\bar{0}\}$
  $F' = F = \times\mathcal{R}$
- $(q, \alpha) \xrightarrow{\quad a \quad} (r, \beta)$ *ssi* $\left\{ \begin{array}{l} q \xrightarrow{\quad a/\phi/\rho \quad} r \\ \exists \gamma \in \mathcal{R} \left\{ \begin{array}{l} \phi \text{ vrai sur } \gamma \\ Reset(\gamma, \rho) = \beta \end{array} \right. \end{array} \right.$
- Reconnaît exactement le langage dé-temporisé qui est donc rationnel

Références

[Alur and Dill, 1994] Alur, R. and Dill, D. L. (1994). A Theory of timed automata. *Theoretical Computer Science*, 126(2):183–235.

[Asarin et al., 2002] Asarin, E., Caspi, P., and Maler, O. (2002). Timed regular expressions. *Journal of the ACM*, 49(2):172–206.

[Bérard and Picaronny, 2000] Bérard, B. and Picaronny, C. (2000). Accepting zeno words: a way towards timed refi nements. *Acta Informatica*, 37(1):45–81.

[Bouyer and Petit, 2002] Bouyer, P. and Petit, A. (2002). A Kleene/Büchi-like theorem for clock languages. *Journal of Automata, Languages and Combinatorics*. To appear.