IFT-2002

Informatique Théorique

H14 - cours 2

Julien Marcil - julien.marcil@ift.ulaval.ca

Langage

Définition: Un **langage** sur un alphabet Σ est un sous-ensemble de l'ensemble Σ^* .

Langage

Considérons les machines M dont les entrées sont un mot d'un alphabet Σ et dont la sortie est 0 ou 1.

$$\begin{array}{ccccc} \text{Entrée} & \longrightarrow & \boxed{M} & & \longrightarrow \text{ Sortie} \\ \Sigma^* & \longrightarrow & f : \Sigma^* \to \{0,1\} & & \longrightarrow \{0,1\} \end{array}$$

On peut aussi dire que la machine **accepte** ou **rejette** son entrée.

Langage d'une machine

Définition: L'ensemble des mots acceptés par la machine M est le **langage de M** que l'on notera $L(M)$

$$L(M) \in \mathcal{P}(\Sigma^*)$$

Théorème

Il existe une fonction que votre langage de programmation favori ne peut calculer.

Démonstration

- On va montrer d'une part que le nombre de programmes est infini et dénombrable.
- Ensuite que le nombre de fonctions $f : \mathbb{N} \to \{0, 1\}$ est non-dénombrable.

Conclusion

L’ensemble des programmes que l’on peut écrire est dénombrable. Donc, il existe un nombre infini non-dénombrable de fonctions pour lesquelles il n’existe aucun programme capable de les calculer.

Automate fini déterministe

Définition: Un **automate fini déterministe** consiste en un quintuple de la forme $(S, \Sigma, \delta, \iota, F)$ où

- S est un *ensemble fini d'états*.
- Σ est *l'alphabet*.
- $\delta : S \times \Sigma \to S$ est la *fonction de transition*.
- $\iota \in S$ est l'*état initial*.
- $F \subseteq S$ est l'ensemble des *états finaux* (ou *accepteurs* ou *acceptants*).

Aujourd'hui

- Langage régulier
- Automates finis non déterministes

Langage régulier

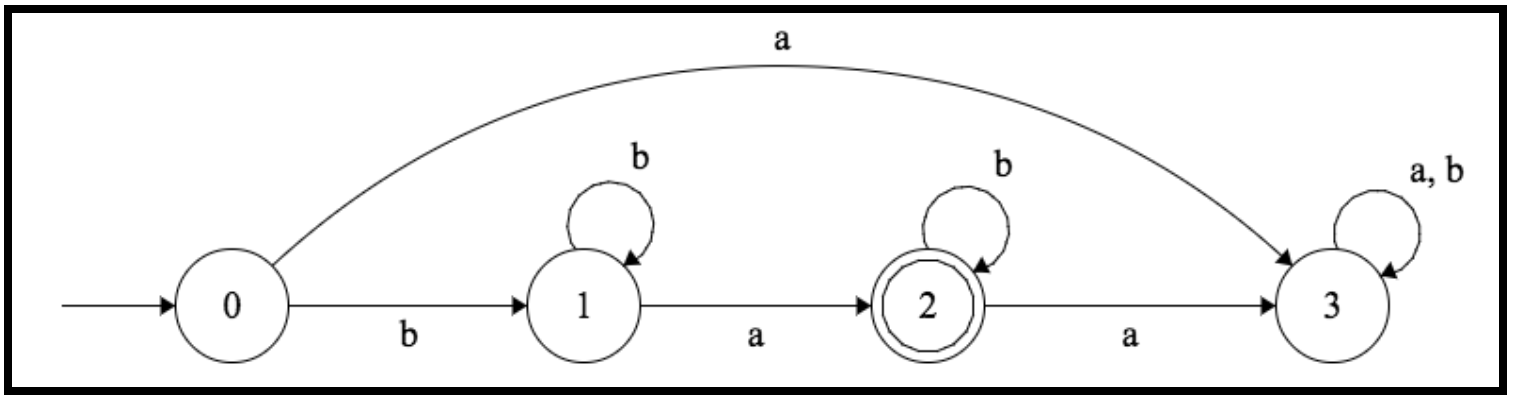
Langage régulier

Définition: Un langage est dit **régulier** s'il existe un automate fini déterministe qui le reconnaît.

Une façon de montrer qu'un langage est régulier est de construire un automate qui reconnaît ce langage.

Example

Soit l'automate fini déterministe M suivant.



Quel est le langage reconnu par l'automate fini déterministe M ?

Les séquences acceptées ont un ou plusieurs b, suivis d'un seul a, suivi de zéro ou plusieurs b. Donc,

$$L(M) = \{b^m a b^n \mid m \in \mathbb{N}^+, n \in \mathbb{N}\}$$

Ce langage est donc régulier.

Exercice

Quel est le langage reconnu par l'automate fini déterministe suivant?

Exercice

Soit l'alphabet $\Sigma = \{b, h, i, o, u\}$. Construire un automate fini déterministe qui accepte le langage $L = \{hibou, hi, bou, hou\}$.

Exercice

Soit l'alphabet $\Sigma = \{a, b\}$. Construire un automate fini déterministe qui accepte le langage $L = \{xa \mid x \in \{a, b\}^*\}$.

Théorème

Le complément d'un langage régulier (relativement à Σ^*) est régulier.

$$L \text{ régulier} \Rightarrow (\Sigma^* - L) \text{ régulier}$$

Démonstration

Montrez comment on peut transformer un automate fini déterministe $M = (S, \Sigma, \delta, \iota, F)$ pour qu' il accepte $\Sigma^* - L(M)$.

Puisque M est déterministe, une séquence d'entrée conduit M à un état et un seul.

La transformation à faire est de rendre finaux les états qui ne l'étaient pas et de rendre non finaux ceux qui l'étaient. Donc

$$L((S, \Sigma, \delta, \iota, S - F)) = \Sigma^* - L(M)$$

Les séquences qui étaient auparavant acceptées sont maintenant rejetées et celles qui étaient rejetées sont acceptées.

Opérations

Soit les languages A et B. Nous définissons les opérations suivantes:

- **Union** : $A \cup B = \{x \mid x \in A \text{ ou } x \in B\}$
- **Çoncaténation** : $A \circ B = \{xy \mid x \in A,\ y \in B\}$
- **Étoile** : $A^* = \{x_1 x_2 \ldots x_k \mid k \geq 0,\ x_i \in A\}$

Example d'union

Pour $\Sigma = \{a, b\}$, soit $L_1 = \{aa, bb\}$ et $L_2 = \{\lambda, aba, bab\}$.

$$L_1 \cup L_2 = \{\lambda, aa, bb, aba, bab\}$$

Example de concatenation

Pour $\Sigma = \{a, b\}$, soit $L_1 = \{aa, bb\}$ et $L_2 = \{\lambda, aba, bab\}$.

$$L_1 \circ L_2 = \{aa, bb, aaaba, aabab, bbaba, bbbab\}$$

Example d'opérateur *

Pour $\Sigma = \{a, b\}$, soit $L = \{aa, bb\}$.

$$L^* = \{\lambda, aa, bb, aaaa, aabb, bbaa, bbbb, aaaaaa, \ldots\}$$

Théorème

L'ensemble des languages réguliers est fermé sur les opérations: *Union*, *Concaténation* et *Étoile*.

Automates finis non déterministes

Diagramme déterministe

Définition: Un diagramme de transitions avec un alphabet Σ est dit **déterministe** si il est *complètement défini* et *non ambigu*.

Diagramme complètement défini

Définition: Un diagramme de transitions avec un alphabet Σ est dit **complètement défini** si, pour chaque symbole $s \in \Sigma$ et chaque état e, il y a au moins une transition étiquetée s qui quitte e.

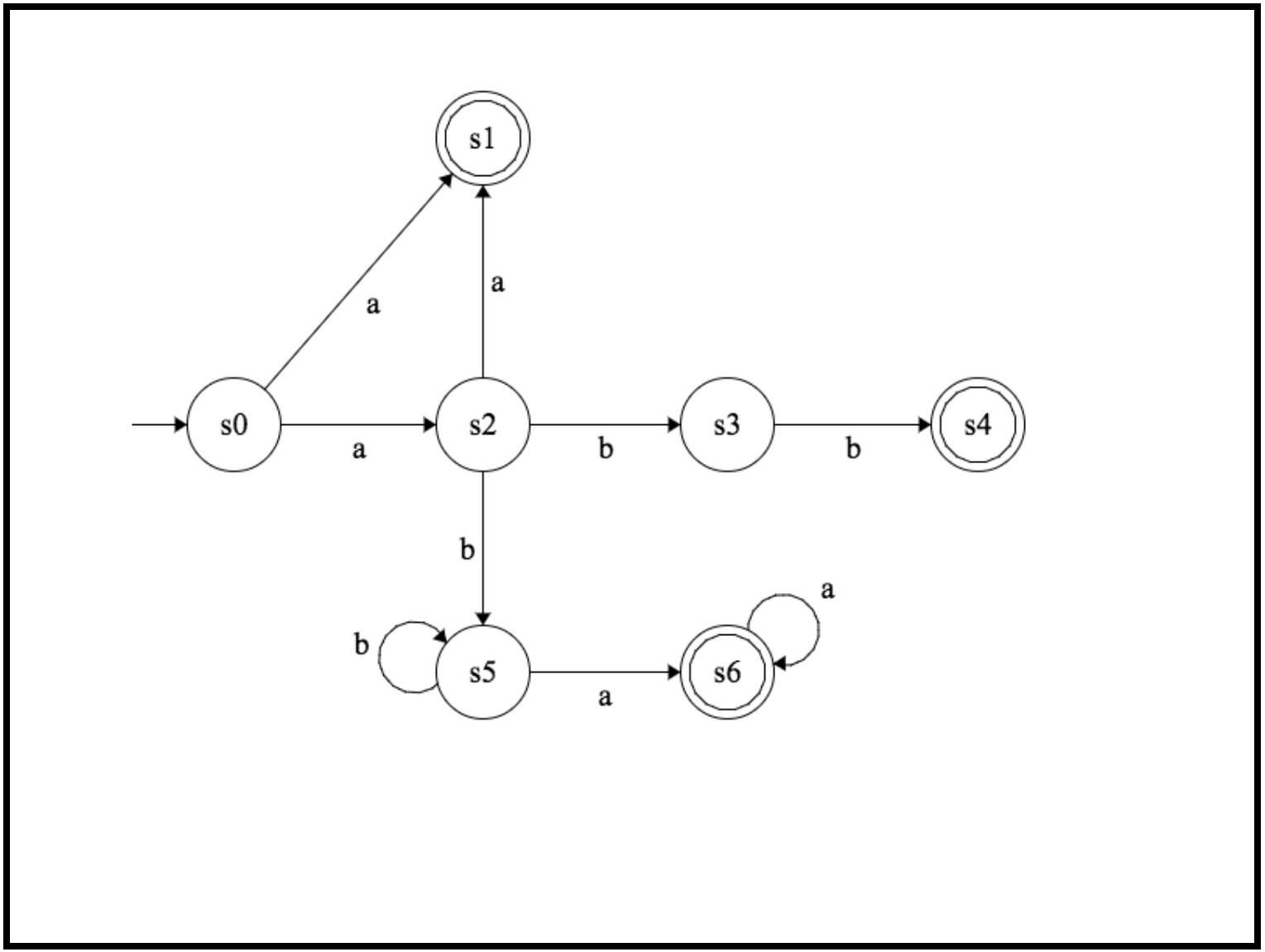
Diagramme non ambigu

Définition: Un diagramme de transitions avec un alphabet Σ est dit **non ambigu** si, pour chaque état et chaque symbole $s \in S$, il existe au plus une transition quittant e et étiquetée s.

Diagramme non déterministes

Nous allons lever deux restrictions que nous avons imposées aux automates déterministes: celle d'être *complètement définis* et celle d'être *non ambigus*.

Exemple



Automate fini non déterministe

Définition: Un **automate fini non déterministe** consiste en un quintuple de la forme $(S, \Sigma, \delta, \iota, F)$ où

- S est un *ensemble fini d'états*.
- Σ est *l'alphabet*.
- $\delta : S \times \Sigma \to \mathcal{P}(S)$ est la *fonction de transition*.
- $\iota \in S$ est l'*état initial*.
- $F \subseteq S$ est l'ensemble des *états finaux* (ou *accepteurs* ou *acceptants*).

fonction de transition δ

Dans les notes de cours, δ est un sous-ensemble de $S \times \Sigma \times S$.

Un élément de δ est un triplet $(s, a, t) \in S \times \Sigma \times S$ où t est un état accessible à partir de l'état s à la lecture d'un a.

Exercice

Donner l'automate non déterministe M correspondant au diagramme de transition ci-dessous.

Exercice

Donner le diagramme de transitions de l'automate suivant:

$$(\{A, B, C, D, E, F, G, H, I\}, \{0, 1\}, \delta, A, \{G, I\})$$

$$\begin{aligned}\delta = \{\ & (A, 1, B), (A, 1, C), (B, 0, E), (C, 0, B), \\ & (C, 0, D), (C, 1, F), (D, 0, A), (D, 0, G), \\ & (D, 1, C), (D, 1, D), (E, 0, H), (E, 1, I), \\ & (F, 1, G), (F, 1, I), (G, 0, F), (G, 1, G), \\ & (H, 0, H), (I, 1, H)\ \}\end{aligned}$$

M accepte la la séquence x

Définition: L'automate fini non déterministe $M = (S, \Sigma, \delta, \iota, F)$ **accepte** (ou *reconnait*) la séquence $x = x_1 x_2 x_3 \ldots x_n$ (où $s_i \in \Sigma$) si et seulement si il *existe* une séquence d'états $s_0, s_1, s_2, \ldots, s_n$ (où $s_i \in S$) tels que

$$\iota = s_0$$

et

$$\forall_{j=1,\ldots,n} \; s_j \in \delta(s_{j-1}, x_j)$$

et

$$s_n \in F$$

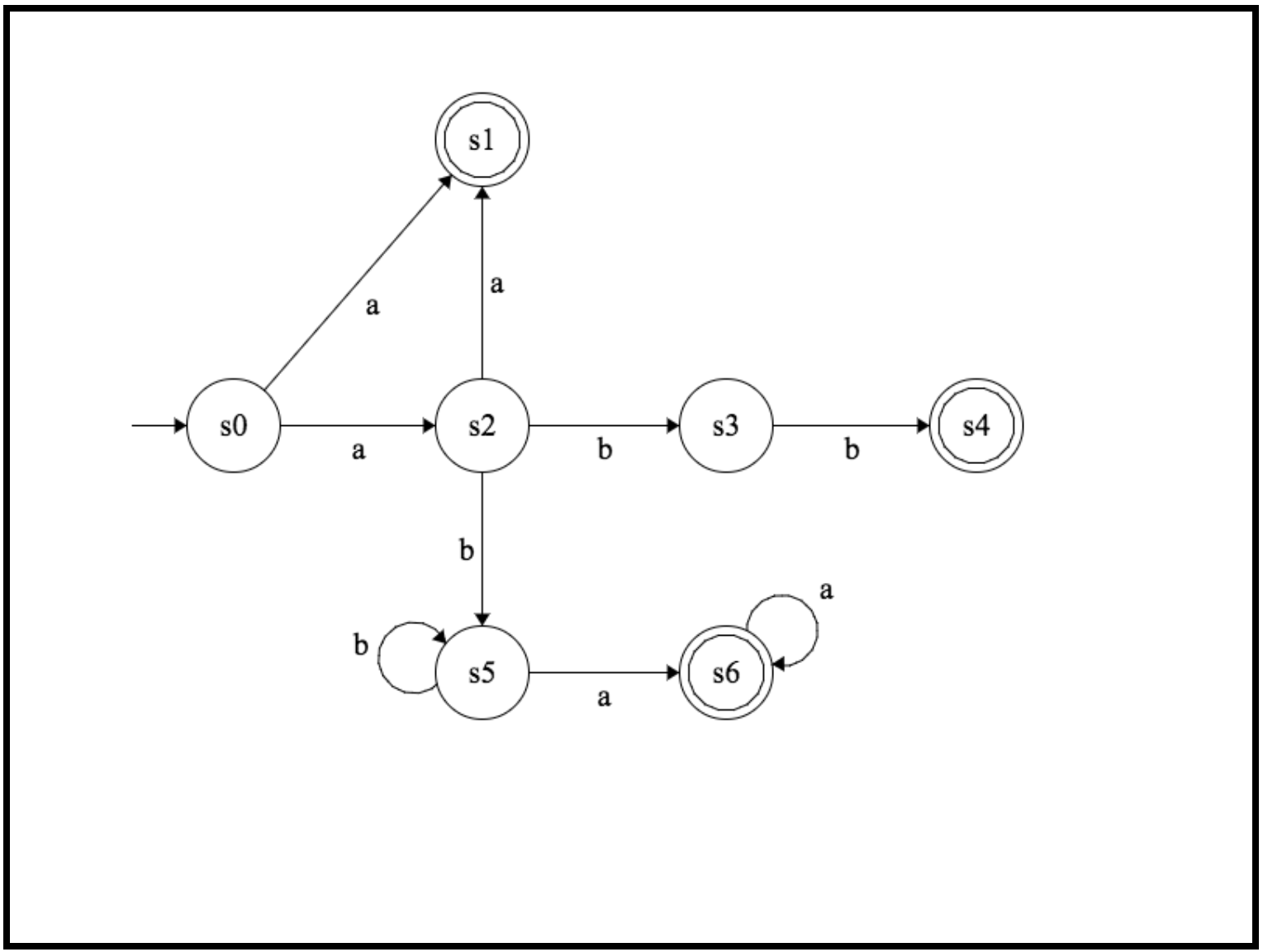
Dans le cas contraire, on dit que l'automate **rejette** la séquence.

M accepte la la séquence x

Une séquence est acceptée s'il est *possible*, en partant de l'état initial, d'atteindre un état final en lisant la séquence; même s'il y a, en plus, des chemins qui mènent à des états non finaux, la séquence est acceptée.

C'est la même notion d'acceptation que pour les automates finis déterministes. Sauf que pour un automate non déterministe on peut trouver plusieurs chemins avec la même séquence (à cause de l'ambiguïté).

Example d'acceptation



Example d'acceptation