Grammaires formelles : Expressions régulières (ou rationnelles) et automates

Karën Fort

karen.fort@sorbonne-universite.fr / https://members.loria.fr/KFort/

Quelques sources d'inspiration

par ordre d'importance décroissant

- *Introduction à la calculabilité* (Pierre Wolper) – InterEditions, 1991
- (excellent) cours en ligne de J-F. Perrot (Paris 6), avec son accord : http://pagesperso-systeme.lip6.fr/Jean-Francois.Perrot/inalco/Automates/Cours4.html
- B. Habert et ses exemples d'automates
- Wikipédia : http://fr.wikipedia.org/wiki/Automate_fini

Langages et alphabets

{aab,aaa,ε,a,b,abababababababbbbbbb}, {ε,aaaaaaaa,a,bbbbbbb} et $\emptyset$ sont des langages de l'alphabet {a,b}

Pour l'alphabet {0,1}, {0,1,00,01,10,11,000,010,000,...} est un langage (le langage contenant tous les mots).

Calculer avec les langages

Nous avons vu :

- les opérations
- les lois

$\rightarrow$ on peut maintenant calculer avec les langages et obtenir de nouveaux langages

$\rightarrow$ les expressions algébriques décrivent des calculs sur des nombres, certains calculs sur les langages peuvent être décrits sous la forme d'expressions

$\rightarrow$ l'expression représente un calcul $\Rightarrow$ on peut lui associer le résultat de ce calcul (valeur de l'expression) = langage (régulier)

Expression régulière

Définition

Dans une expression régulière :

- n'interviennent que les opérations $\cup$, * et produit
- le calcul commence à partir de singletons d'une seule lettre

Les langages qui peuvent être décrits par des expressions régulières sont des langages réguliers

Langages réguliers

L'ensemble R des langages réguliers sur un alphabet $\sum$ est le plus petit ensemble de langages satisfaisant les conditions suivantes :

1. $\emptyset \in R$; $\varepsilon \in R$
2. $\{a\} \in R$ pour tout $a \in \sum$
3. si $A, B \in R$, alors $A \cup B, AB$ et $A* \in R$

 il existe des langages non réguliers (beaucoup plus que des réguliers)

 tous les langages finis sont réguliers (tous les mots du langage sont énumérables)

Exemple :
le langage $L = \{ab, raca, dabra\}$ est la valeur de l'expression
$E = \{a\}\{b\} \cup \{r\}\{a\}\{c\}\{a\} \cup \{d\}\{a\}\{b\}\{r\}\{a\}$

En programmation, on écrit :
$E = ab|raca|dabra$

Pour quoi faire ?

Définir en termes finis un ensemble infini : { Pierre, Paul, Jacques, Jules, Michel, Nestor, . . . } ou {0, 2, 4, 6, 8, . . . }

- . . . note l'infini : perception intuitive de la récurrence entre les premiers termes
- impossible à déduire pour les machines
- il faut une définition explicite et exhaustive, formulée par un objet fini

→ un programme (qui est un texte fini) peut être vu comme la définition de l'ensemble infini des séquences de calculs auxquelles son exécution pourra donner naissance

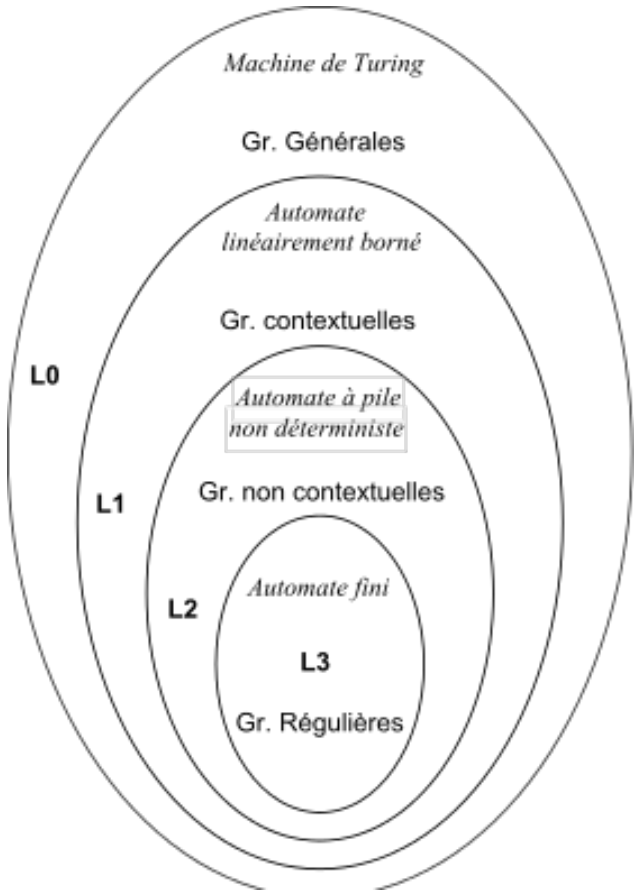
Cas des langages

On définit l'infini grâce à deux classes de procédés :

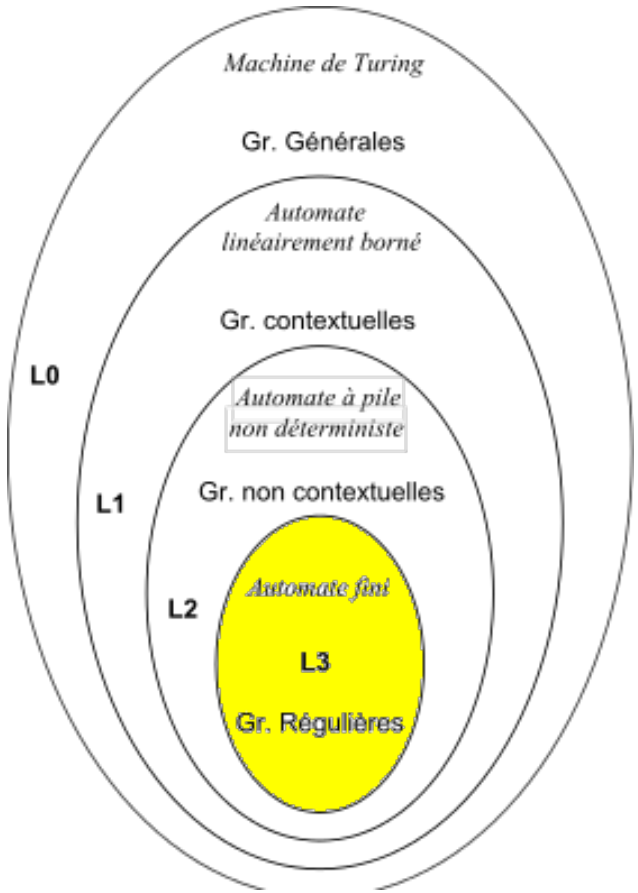
1. les grammaires :
   - au sens de la grammaire générative (Chomsky)
   - ensemble (fini) de règles de construction qui assurent que tout mot produit en les utilisant fait partie du langage
   - réciproquement : tout mot du langage peut être construit par ces règles
2. les automates :
   - permettent de décider si un mot appartient au langage

La théorie des langages établit une correspondance entre classes de langages, classes de grammaires et classes d'automates

Classes de langages, classes de grammaires et classes d'automates



Langages réguliers, grammaires régulières, automates finis



Calcul des langages, valeur

une expression régulière représente un calcul sur les langages, et ce calcul aboutit à un langage qui est la valeur de l'expression

Exemple : $yx^*xy^*x = yxxx^* \cup yxyy^*x \cup yxxx^*yy^*x$

- les deux expressions sont différentes
- leur valeur est égale

dès qu'une expression régulière a pour valeur un langage infini (dès qu'elle contient *), nous n'avons aucun moyen de désigner directement le langage

$\Rightarrow$ besoin d'une désignation unique ($\neq$ expressions régulières) pour un langage donné

Lois du calcul sur les langages

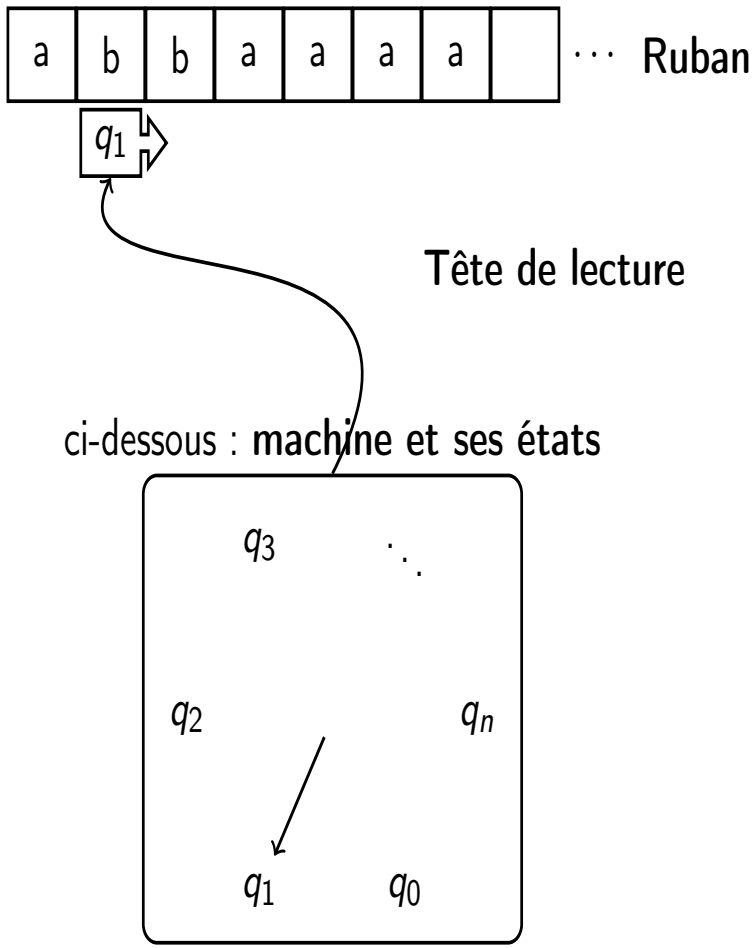
- pour tout langage L, on a
  $LL^* = L^*L = L^+$
- pour tous langages A et B, on a
  $(A \cup B)^* = (A^*B)^*A^* = A^*(BA^*)^*$

Automate fini (*finite-state automaton*)

- représentation d'un algorithme destiné à associer une valeur booléenne (vrai ou faux) à chaque mot sur un alphabet X
- l'ensemble de ces mots constitue le langage reconnu par l'automate

- l'automate est dit « fini » car il possède un nombre fini d'états : il ne dispose donc que d'une mémoire bornée

Mécanisation



- le mot à traiter est lu de gauche à droite (ici)
- la valeur *vrai* ou *faux* est obtenue à la fin du programme
- après la lecture de chaque lettre, l'automate est dans un certain état qui matérialise l'information emmagasinée au cours de la lecture des lettres précédentes
- la lecture de la prochaine lettre fait passer l'automate dans un autre état : c'est une transition
- si le calcul s'arrête dans un état final (terminal), alors le résultat est vrai

Automate fini sur un alphabet X

Définition

un automate fini sur un alphabet X est constitué par :

- un ensemble fini dont les éléments sont appelés états
- une fonction de transitions qui, à chaque état et à chaque lettre de l'alphabet, associe un état
- un état initial
- un ensemble d'états terminaux (finaux)

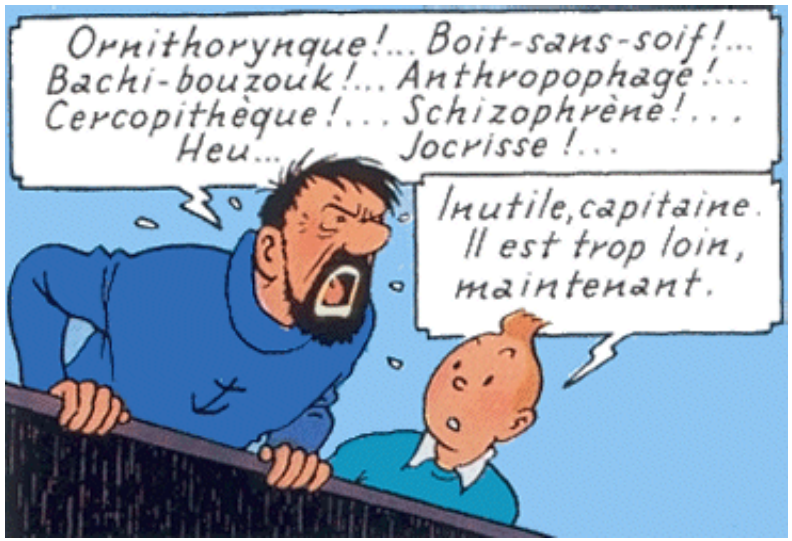
Codes

- de portes : digicode
- pour obtenir une boisson dans une machine

Livre dont vous êtes le héros

Un conte à votre façon , Queneau, Raymond [1967], in Oulipo. La littérature potentielle (Créations Re-créations Récréations). Paris : Gallimard, coll. «Folio Essais», 1973, pp.273-276.

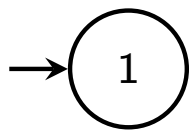
Les injures du Capitaine Haddock



©Hergé

Automate pas à pas

État initial :

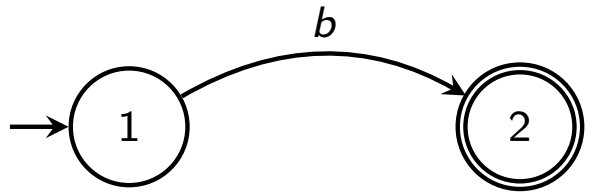


Automate pas à pas

État initial et état final :

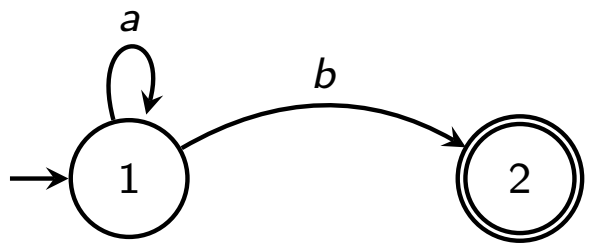
Automate pas à pas

Ajout d'une transition : l'automate reconnaît b



Automate pas à pas

Ajout d'une transition : l'automate reconnaît a^*b



Que reconnaît l'automate suivant ?

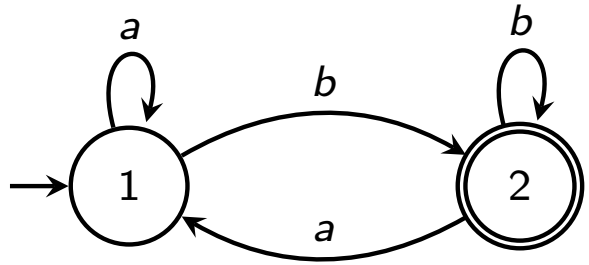
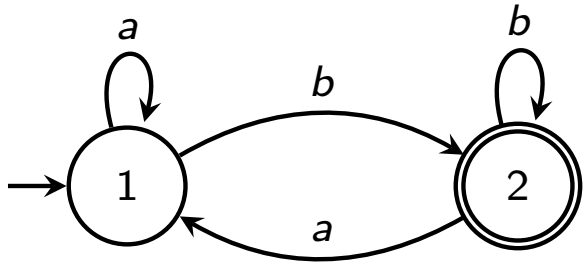
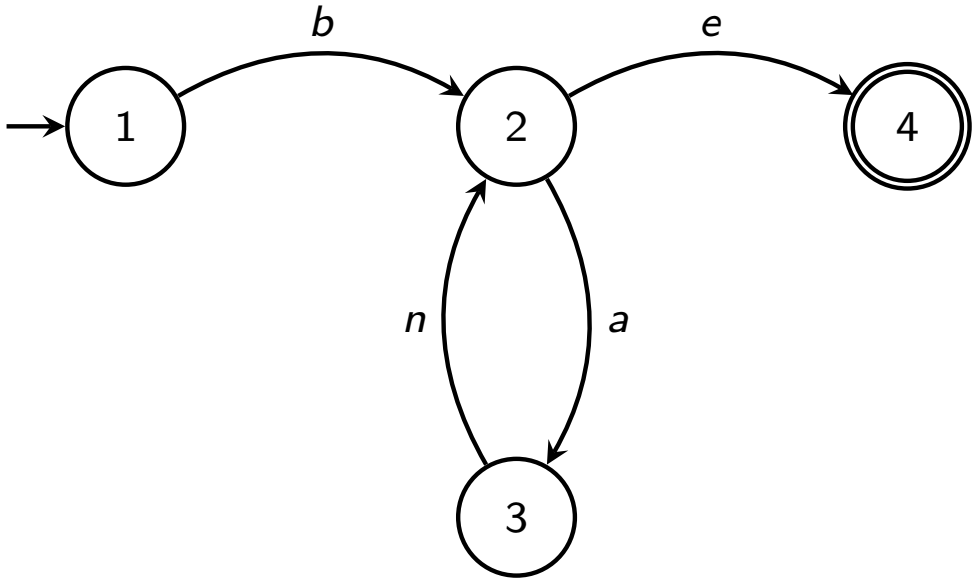


Table de transition de l'automate

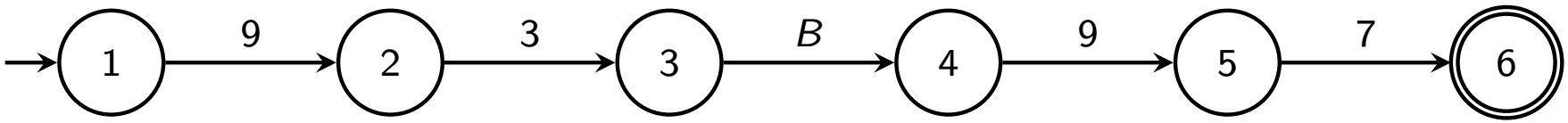


	a	b
1	1	2
2	1	2

Que reconnaît l'automate suivant ?



Digicode : 93B97

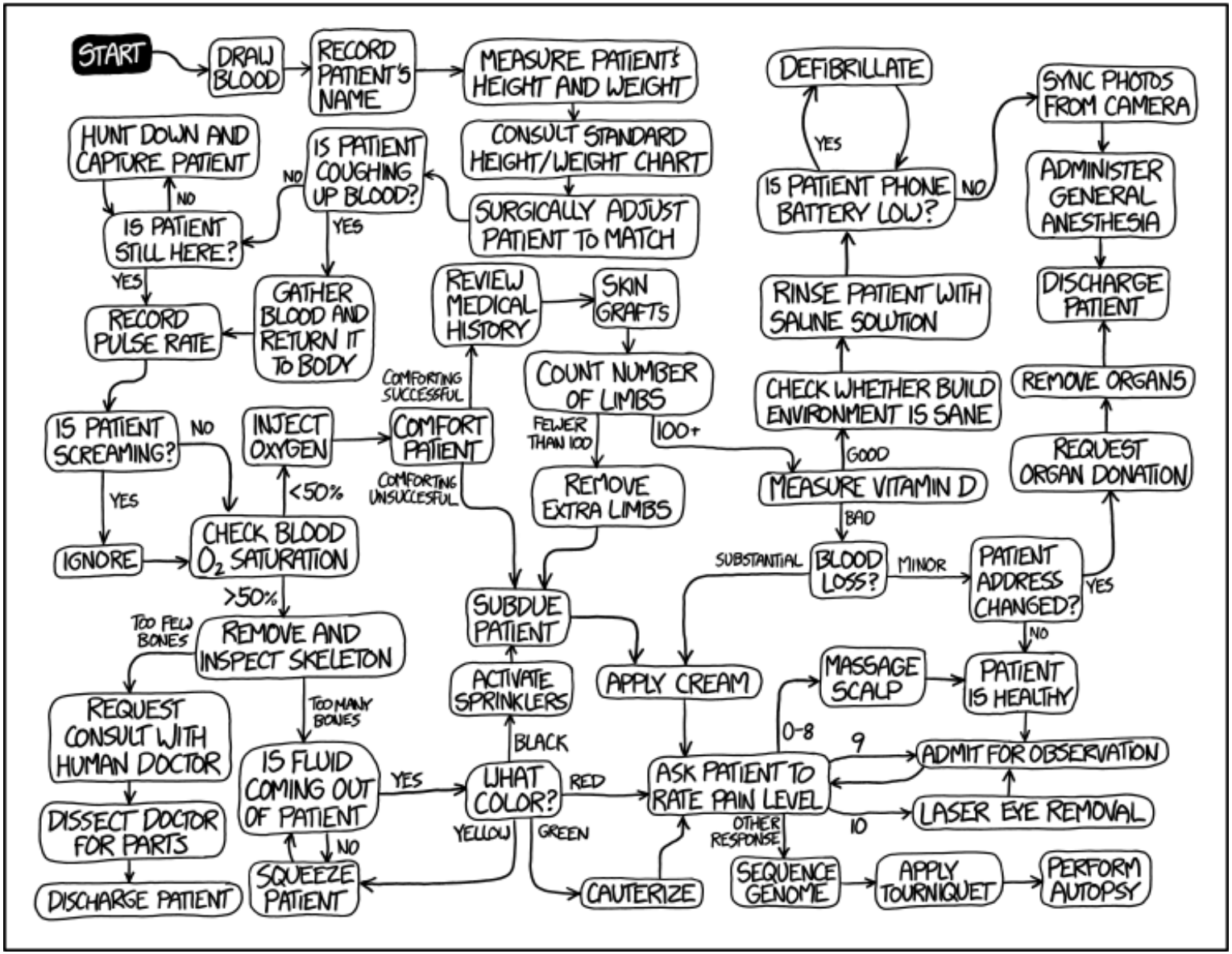


$\rightarrow$ tant que le bon code n'est pas entré

= tant que l'état final n'est pas atteint

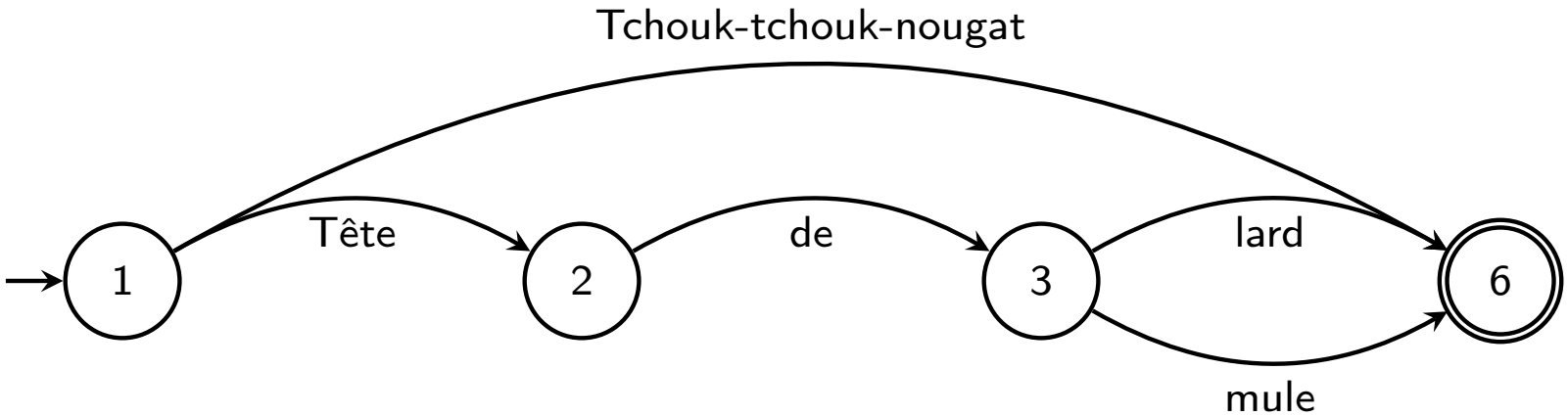
la porte ne s'ouvre pas

A GUIDE TO THE MEDICAL DIAGNOSTIC AND TREATMENT ALGORITHM USED BY IBM's WATSON COMPUTER SYSTEM



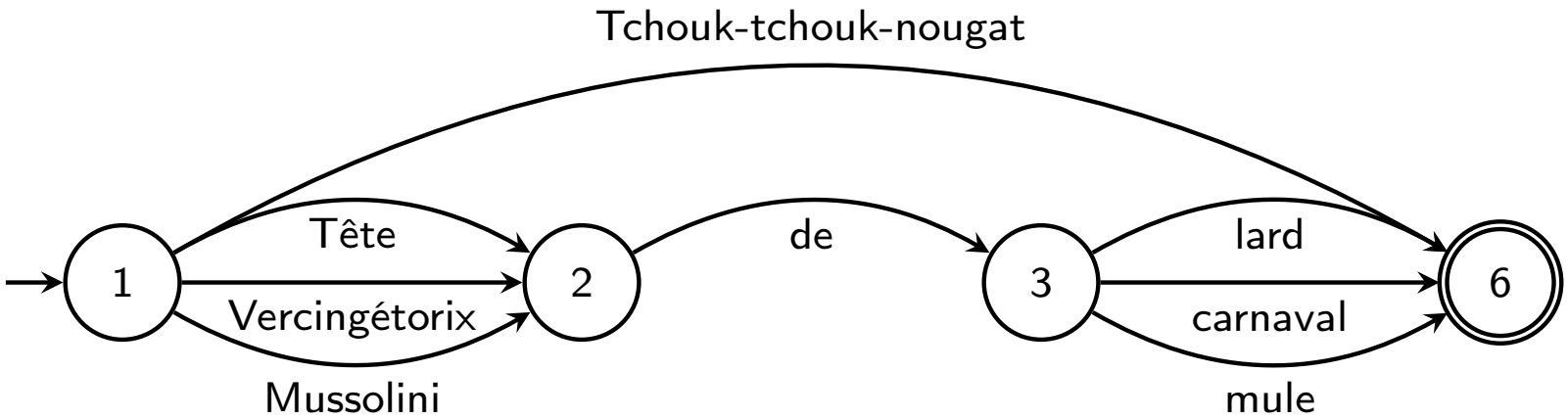
https://xkcd.com/1619/

Les injures du Capitaine Haddock : un (petit) bout



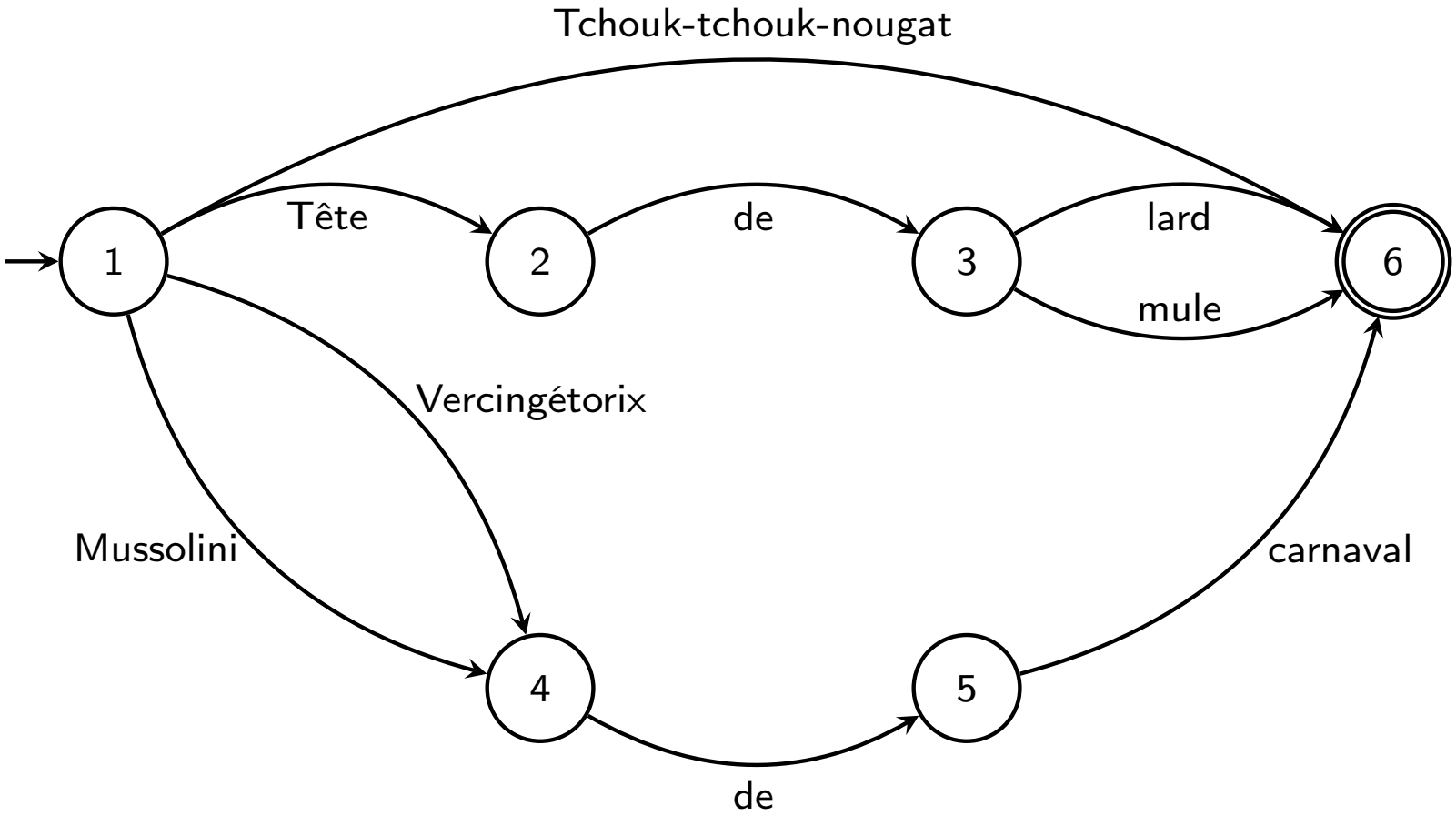
Comment ajouter « Mussolini de carnaval » et « Vercingétorix de carnaval » ?

Les injures du Capitaine Haddock : un (petit) bout



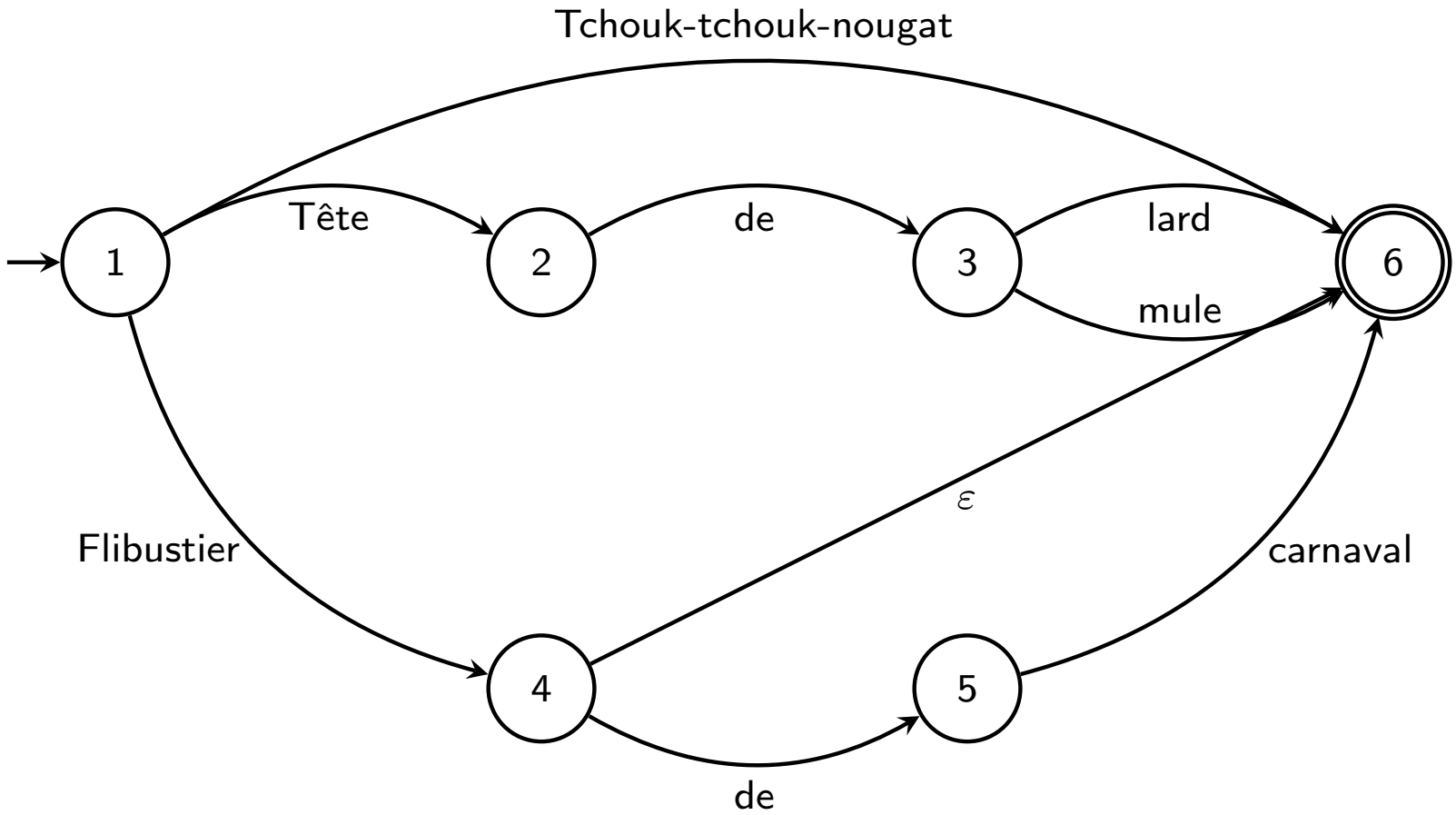
Problème ?

Les injures du Capitaine Haddock : un (petit) bout



Ajoutez « Flibustier » et « Flibustier de carnaval »

Les injures du Capitaine Haddock : un (petit) bout



$\rightarrow$ tout ce qui n'est pas reconnu (accepté) par l'automate n'est pas une insulte du Capitaine Haddock

- une expression régulière a pour valeur un langage (régulier)
- un langage régulier peut correspondre à plusieurs expressions régulières
- un langage régulier peut être représenté par un automate

- expressions régulières
- langage régulier
- automate :
  - définition
  - représentation graphique

Exercice 1 : à faire, ici et maintenant

Construire les automates suivants :

- qui reconnaît le digicode (imaginaire) : 0A753
- qui reconnaît les mots (sur $L = \{a, b\}$) contenant un nombre impair de lettres « a »
- qui reconnaît (sur $L = \{a, b\}$) $a * b * b$
- qui reconnaît les mots (sur $L = \{a, b\}$) commençant par aba
- qui reconnaît les mots (sur $L = \{a, b\}$) finissant par aba

et les tester !

Exercice 2 : à faire, ici et maintenant

On considère l'alphabet usuel des caractères utilisés pour écrire un texte en français. Construire l'automate reconnaissant :

- les dates écrites dans un format illustré par l'exemple 04/05/2004 (on suppose que le mois de février peut avoir 29 jours sans vérification des années bissextiles)
- les formes fléchies du verbe « finir » à l'indicatif présent

Exercice 3 : à faire, ici et maintenant

Construire sur papier (ou sur le simulateur, avec des correspondances) l'automate qui reconnaît les insultes du Capitaine Haddock (lettre B, voir Wikipédia).

Exercice 4 : à faire, ici et maintenant

Construire l'automate qui reconnaît un numéro de sécurité sociale de Français nés en métropole.
On considérera pour l'exercice que la clé de contrôle est un nombre à deux chiffres aléatoire.

Vous pouvez bien sûr utiliser : `http://fr.wikipedia.org`