

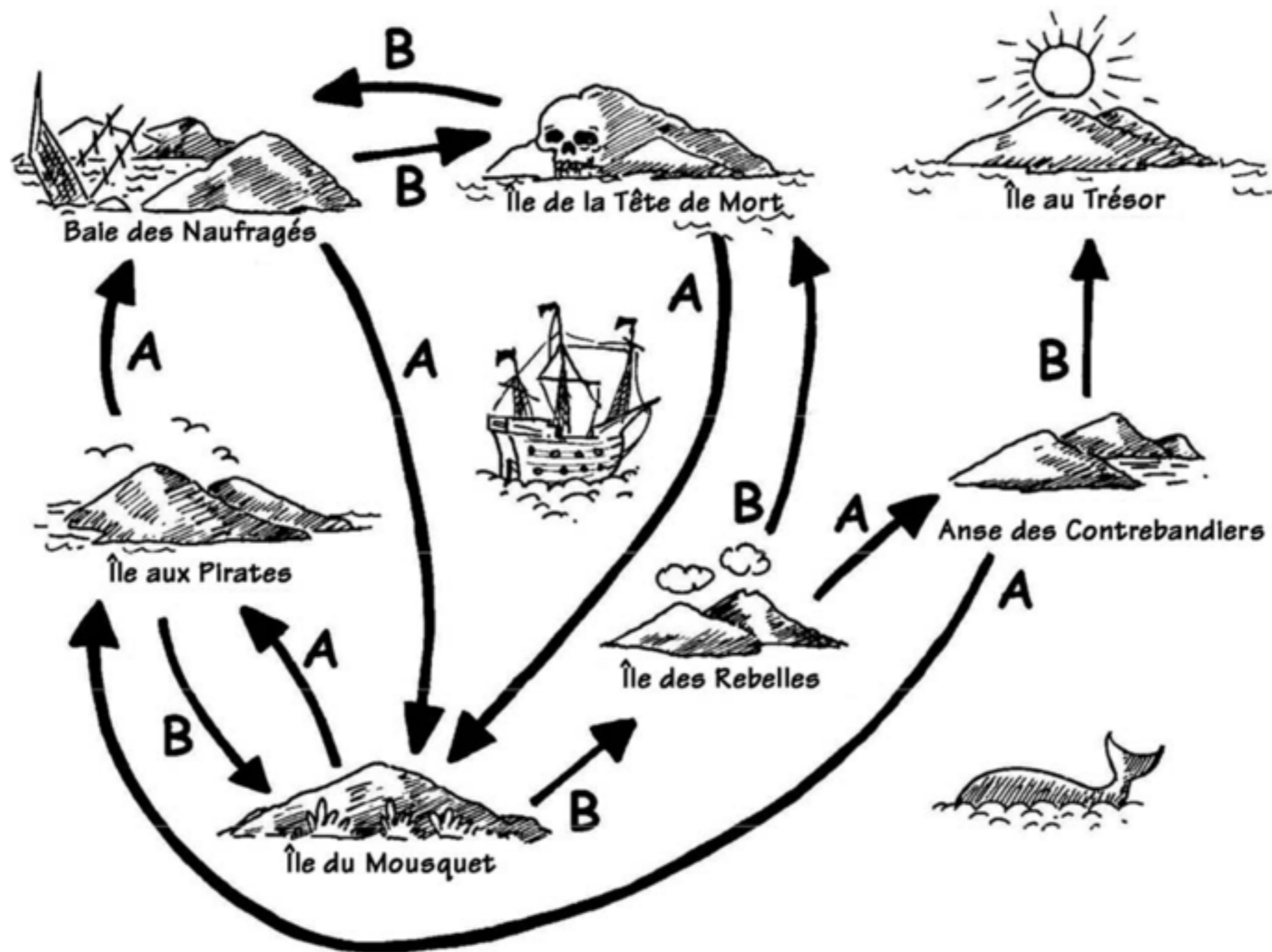
# Les automates

Fabrice EUDES, Pascal EVRARD, Philippe MARQUET,  
François RECHER & Yann SECQ

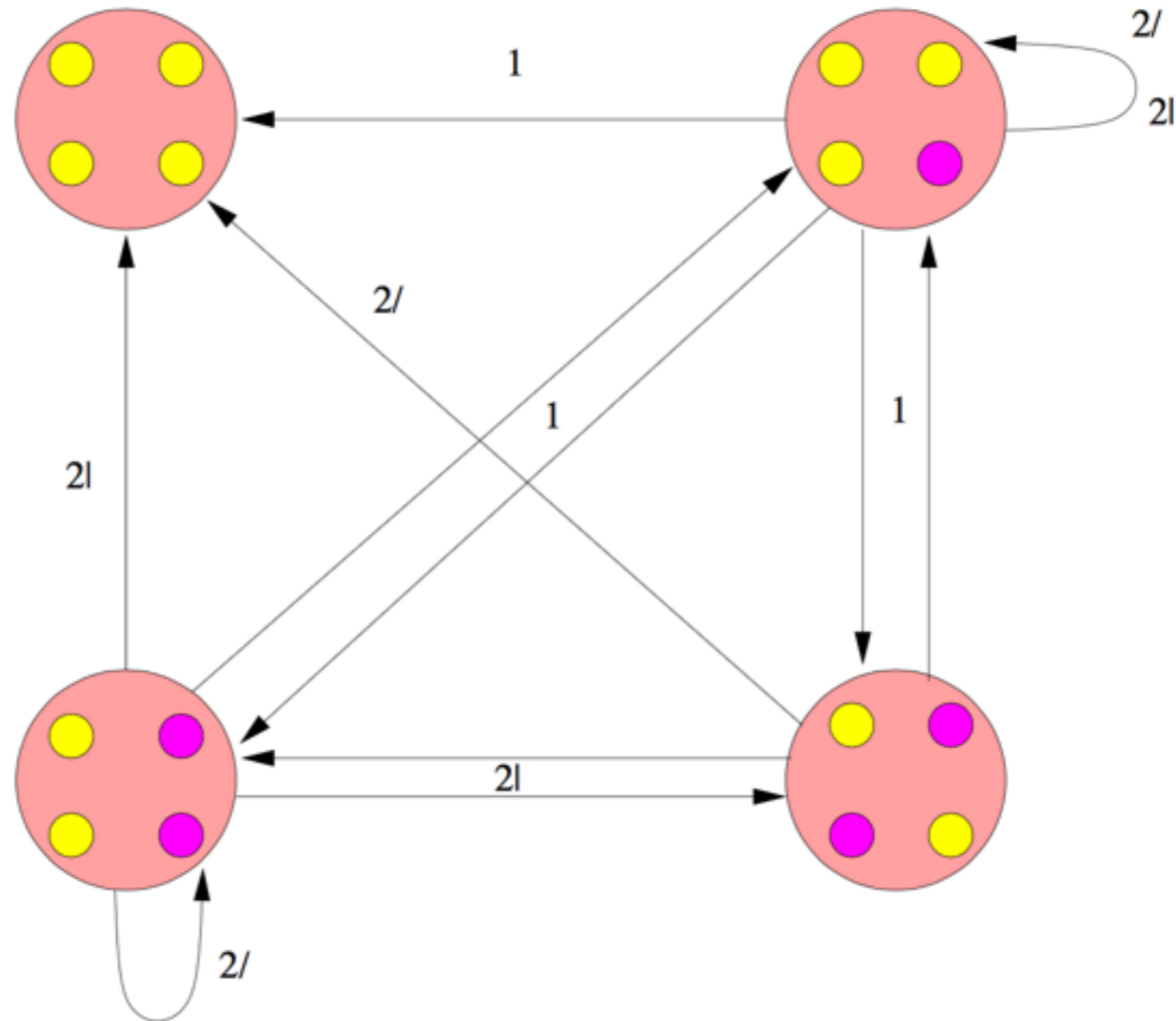
# Retour sur l'île et le barman

- Deux problèmes similaires:
  - Des îles, des bateaux et un trésor à trouver
  - Des verres dans différentes positions et une situation particulière à atteindre
- A chaque fois, on manipule des concepts similaires:
  - un ensemble de **noeuds** (les îles et les états des positions des verres),
  - des **arcs** (les bateaux et les actions de retournement des verres)

# L'île au trésor



# Le barman aveugle avec des gants de boxe



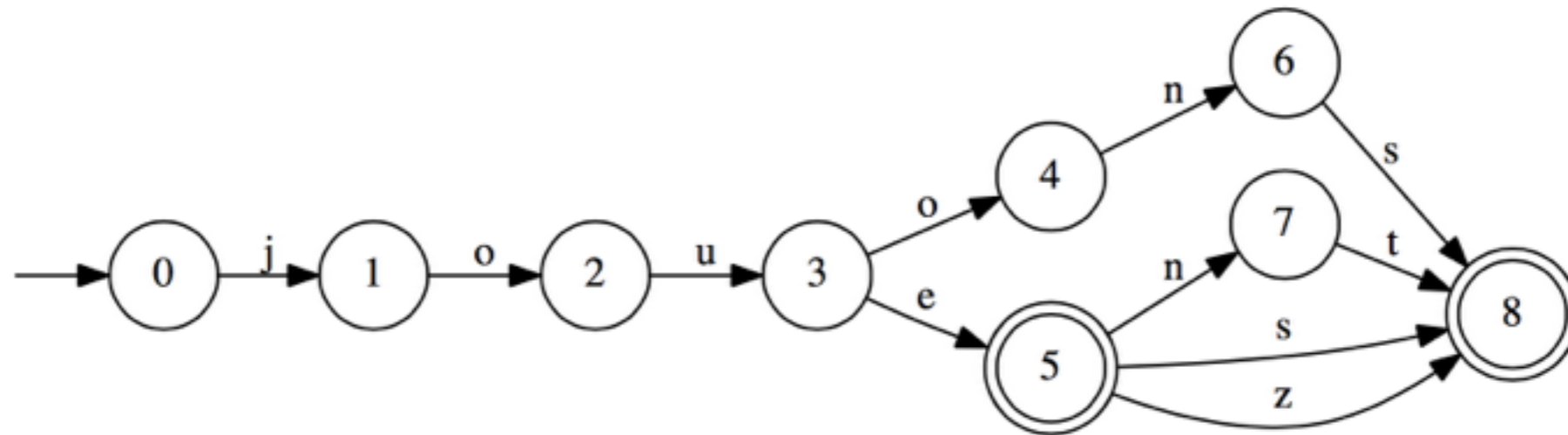
# Les automates

- Ensemble “d’états du système”, reliés par des “transitions” marquées par des symboles
- Étant donné un “mot” en entrée, l’automate lit les symboles du mot un par un et va d’état en état selon les transitions. Le mot lu est soit accepté par l’automate soit rejeté.
- Quelques exemples classiques d’utilisation d’automates :
  - Vérification d’un circuit électronique, recherche d’occurrence dans un texte
  - Vérification de protocoles de communication, compression de données
  - Compilation, ...
- Comprendre ce qu’un ordinateur peut faire (décidabilité) et ce qu’il sait faire efficacement (complexité)

# Plusieurs types d'automates

- Il existe différentes classes d'automates:
  - les automates à états finis
  - les automates à pile
  - les réseaux de Pétri
  - la machine de Turing
- Concept fondamental en informatique

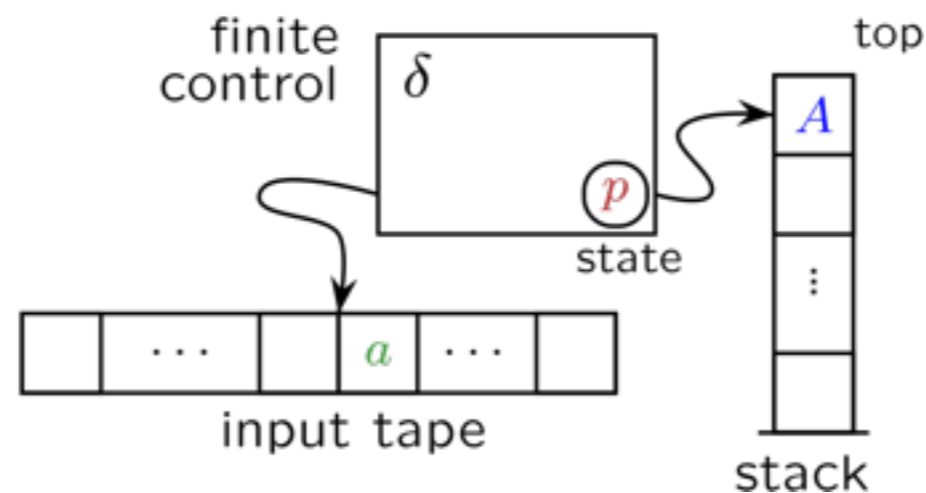
# Les automate à états finis



- Alphabet: {e, j, n, o, s, t, u, z}
- 9 états et 11 transitions
- Un état initial et un ou plusieurs états finaux
- Avec "jouons" en entrée, l'automate accepte le mot
- Avec "jouet" en entrée, l'automate n'accepte pas le mot
- Un automate définit ainsi un langage (cf. Chomsky)

# Les automates à pile

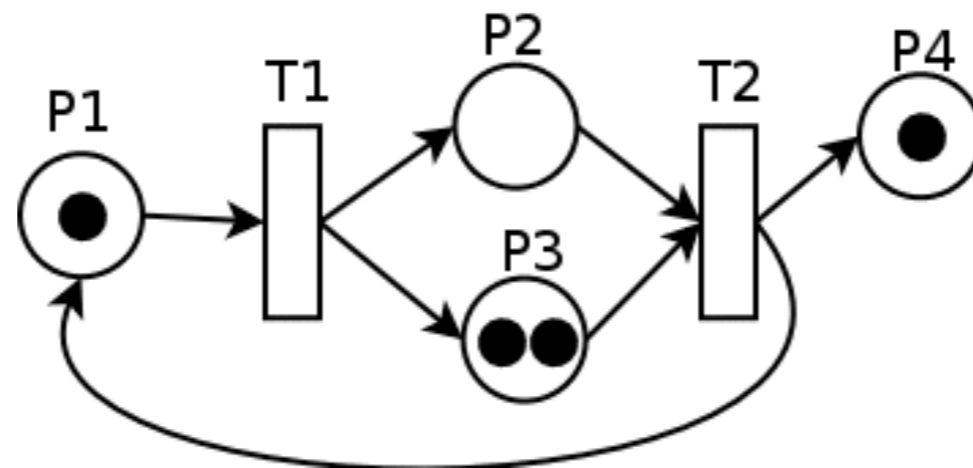
- Ajout d'une mémoire à l'automate sous la forme d'une pile (LIFO)
- Le sommet de la pile est pris en compte en plus de la lettre courante et l'état de l'automate
- Nombreuses applications:
  - analyse syntaxique des langages de programmation
  - transformation d'algorithmes récursif en itératif





# Les réseaux de Pétri

- Réseau de Pétri ou réseau places/transitions
- Modéliser et vérifier le comportement dynamique des systèmes à événements discrets (système de télécommunication, réseaux de transport, ...)
- Des places, des transitions et des jetons
- Représentation graphique sous la forme d'un graphe biparti orienté



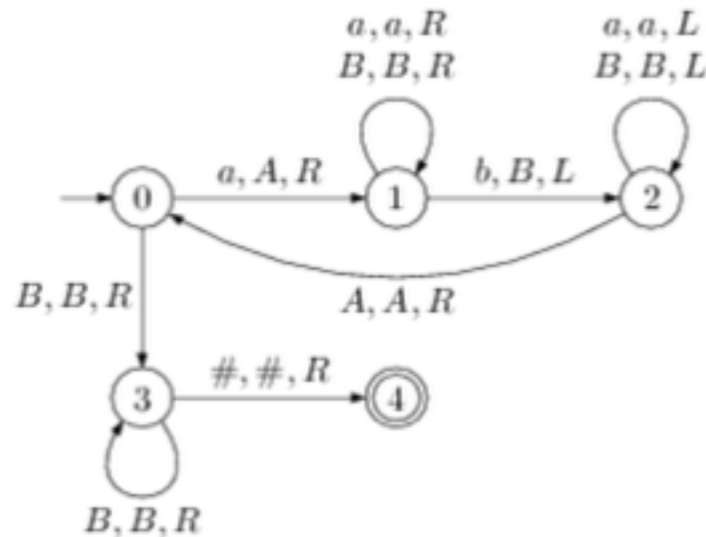
# Problème de la décision de Hilbert et machine de Turing

- *Entscheidungsproblem*: existe-t-il un algorithme qui prend une formule en logique du premier ordre et détermine si oui ou non la formule est universellement valide
- Turing propose l'utilisation d'une « machine » pour répondre négativement à ce problème (Church aussi)
- Permet de définir la notion de calculabilité et décidabilité
- Machine: un ruban infini, une tête de lecture, une table d'état, possibilité de lire et écrire la case en dessous de la tête de lecture

# Une machine de Turing pour le langage $X = \{a^n b^n \mid n \geq 0\}$

Un machine M qui accepte le langage X est la suivante

- l'ensemble des états est  $Q = \{0, 1, 2, 3, 4\}$ ,
- l'alphabet d'entrée est  $\Sigma = \{a, b\}$ ,
- l'alphabet de bande est  $\Gamma = \{a, b, A, B, \#\}$ ,
- l'état initial est  $q_0 = 0$ ,
- le seul état final est l'état 4,
- l'ensemble des transitions est décrit par la figure ou le tableau ci-dessous. Cette machine est *déterministe*.



$\delta$	a	b	A	B	#
0	1,A,R			3,B,R	
1	1,a,R	2,B,L		1,B,R	
2	2,a,L		0,A,R	2,B,L	
3				3,B,R	4,#,R
4					

No	Configuration	No	Configuration	No	Configuration	No	Configuration	No	Configuration
0	0aaabbb	6	2AaaBbb	12	AA2aBBb	18	AAAB2BB	24	AAABBB3#
1	A1aabbb	7	A0aaBbb	13	A2AaBBb	19	AAA2BBB	25	AAABBB#4#
2	Aa1abbb	8	AA1aBbb	14	AA0a1BBb	20	AA2ABBB		
3	Aaa1bbb	9	AAa1Bbb	15	AAA1BBb	21	AAA0BBB		
4	Aa2aBbb	10	AAaB1bb	16	AAAB1Bb	22	AAAB3BB		
5	A2aaBbb	11	AAa2BBb	17	AAABB1b	23	AAABB3B		

Calcul avec l'entrée aaabbb.

source

# Caractéristiques communes

- Un automate:
  - prend en entrée des données discrètes (souvent un mot)
  - possède un nombre fini de configurations internes, appelées état (éventuellement infini)
  - peut disposer d'une mémoire (pile(s), ruban, ...)
  - évolue selon un nombre fini de règles
  - possède une ou des configurations d'acceptation

# Les automates

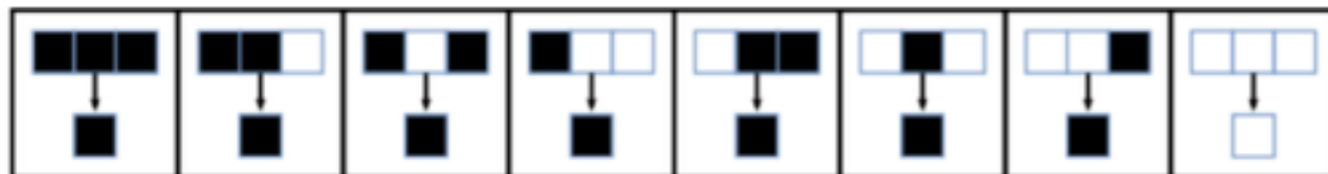
- Un (des) formalisme(s) simple(s) mais fondamental(aux) en informatique
- De nombreux usages en informatique théorique et dans des domaines appliqués
- Illustrable avec l'île au trésor (petits) et le barman (plus grands)
- Et il y a aussi les automates cellulaires ;)

# Quelques pistes

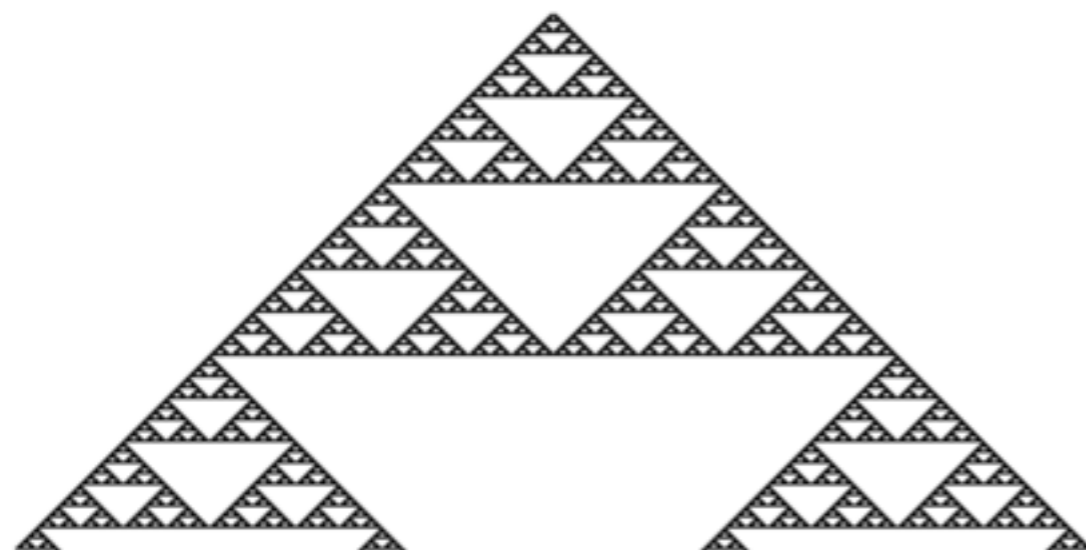
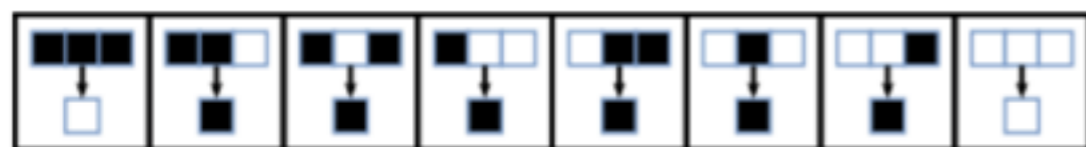
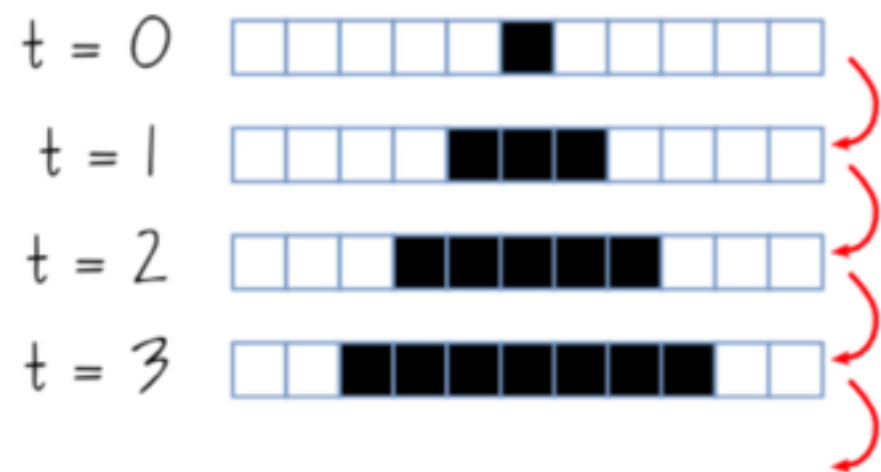
- Automates et Langage (S4/S5 Lic. Info)
- Langages rationnels, automates fini de mots, automates à pile, machine de Turing
- Automates cellulaires de Wolfram
- *à compléter ...*

# Automates cellulaires

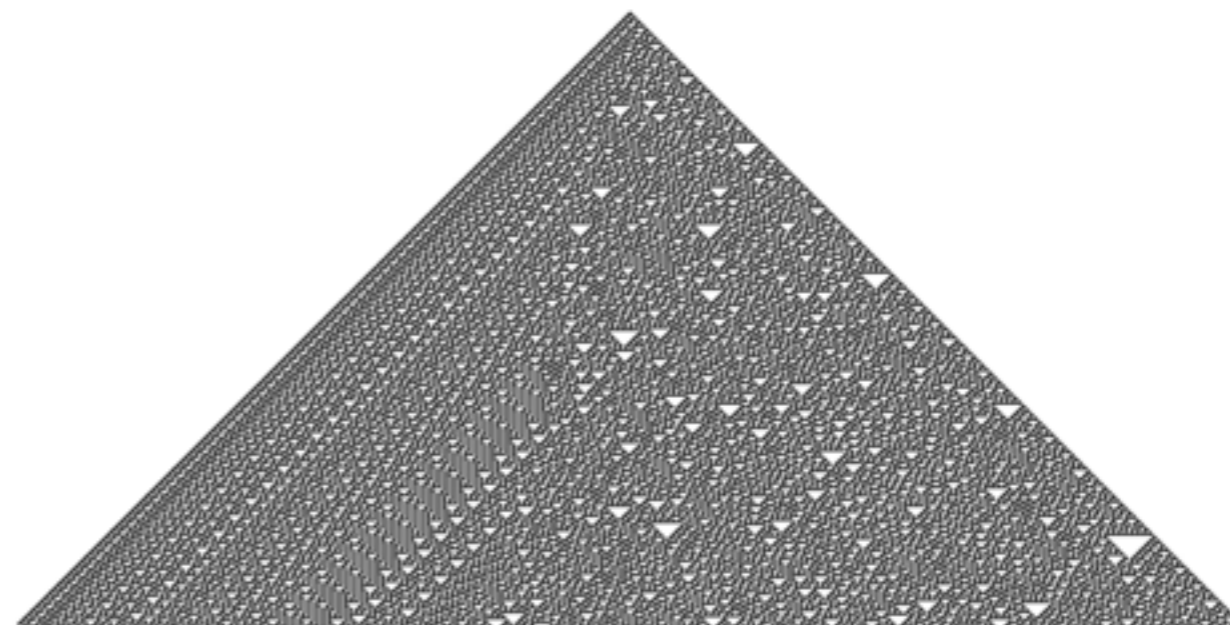
- Une grille (souvent 2D) d'automates
- Modèle standard de l'étude de systèmes complexes
- Comportements locaux => phénomène « émergent »
- Deux exemples:
  - Les automates de Wolfram (une dimension)
  - Le jeu de la vie de Conway (deux dimensions)



*règle 254*



*règle 126*



*règle 30*



# Le jeu de la vie

- Simulation de l'évolution d'une population de bactéries
- Grille 2D + règles simples d'évolution:
  - si vivant, survit avec 2 ou 3 voisins, sinon meurt
  - si morte, naissance si exactement 3 voisins
- Nombreux motifs émergents
- Equivalent à une machine de Turing !

[golly.sourceforge.net](http://golly.sourceforge.net)