Complexité et Algorithmique

J. Bond, bond@polytech.unice.fr

☎ 04 89 15 41 57

D. Mazauric, dorian.mazauric@inria.fr

C. Papazian, papazian@polytech.unice.fr

Le cours en deux parties

- **Le semestre est composé de deux parties**
  - **première partie : Complexité ~ 8 semaines**
  - **seconde partie : Algorithmique ~ 4 semaines**

- **Notation :**
  - **Contrôle écrit partie 1 – coeff 4**
  - **Contrôle écrit partie 2 – coeff 2**
  - **Miniprojet 1 – coeff 3**
  - **Miniprojet 2 – coeff 3**

Exemple de complexité

- un problème de tri selon l'ordre lexicographique de n mots d'un dictionnaire (tous les mots étant de longueur 20 caractères). Notre problème est de taille $20n$.
- On essaye plusieurs méthodes :
  - $2\,n^5$ pour la méthode **A** (très inefficace),
  - $20\,n^2$ pour la méthode **B** (peu efficace),
  - $40\,n\log_2 n$ pour la méthode **C** (assez efficace)
  - $80\,n$ pour la méthode **D** (très efficace, tient compte de la nature des données).

Exemple (suite)

- on suppose, que notre machine est cadencé 100 MHz, c.a.d. peut effectuer 100 000 000 instructions par seconde, alors un problème de taille 20 000 nécessitera
  - plus de 20 294 siècles pour la méthode A,
  - 80 secondes pour la méthode B,
  - 114 mili-secondes pour la méthode C,
  - 16 mili-secondes pour la méthode D.

Exemple (suite)

- Par contre, si on se fixe un temps de calcul de 10 minutes, on peut traiter des problèmes de taille au plus
  - 124 pour la méthode A.
  - 54 772 pour la méthode B,
  - 58 154 442 pour la méthode C,
  - 750 000 000 pour la méthode D.

Exemple (suite)

- on suppose, notre machine (plus réaliste …) dispose d’un processeurs à quatre cœurs cadencés 3GHz, c.à.d. peut effectuer 12 000 000 000 instructions par seconde, alors un problème de taille 50 000 nécessitera
  - plus de 16 515 siècles pour la méthode A,
  - 4 secondes pour la méthode B,
  - 2.6 mili-secondes pour la méthode C,
  - 0,33 mili-secondes pour la méthode D.

Exemple (suite)

- Par contre, si on se fixe un temps de calcul d’une minute, on peut traiter des problèmes de taille au plus
  - 204 pour la méthode A.
  - 189 736 pour la méthode B,
  - 616 450 800 pour la méthode C,
  - 9 000 000 000 pour la méthode D.

Et si les machines …

Complexité	taille actuelle	taille future
$O(n)$	N	$N * 1000$
$O(n^2)$	N	$N * 32$
$O(n^3)$	N	$N * 10$
$O(n^5)$	N	$N * 4$
$O(2^n)$	N	$N + 10$
$O(3^n)$	N	$N + 7$

Conclusion

- Il faut savoir (si possible …) si un problème est oui ou non traitable !

- Difficile, car c'est très difficile de trouver la cmplexité d'un problème (attention non d'un algorithme mais d'un problème) !

L’exemple de Garey & Johnson

Vous êtes le chef du service informatique d'une entreprise.

L'exemple de Garey & Johnson

Un nouveau problème se pose, et on fait appel à vous pour construire un programme résolvant le problème.

L’exemple de Garey & Johnson

Vous vous mettez au travail, mais au bout de quelques jours, voir quelques semaines de travail vous n'avez toujours pas de solution raisonnable au problème, sinon d'essayer toutes les possibilités, ce qui revient à l’uti - lisation d'un algorithme exponentiel, et n'est donc pas réaliste.

L’exemple de Garey & Johnson

Vous êtes conscient que, si votre réponse est négative, votre emploi sera en jeu. Il est donc souhaitable de trouver un autre moyen de s'en sortir.

L’exemple de Garey & Johnson

L'idéal serait de pouvoir montrer qu'il ne peut pas y avoir de solution au problème, mais la preuve en est trop difficile, et vous ne savez pas la faire.

L’exemple de Garey & Johnson

Th. des problèmes NP-complets : permet d'affirmer que, même si vous ne savez pas résoudre le problème, vous licencier ce n'est pas une solution pour l'entreprise, car personne (donc votre remplaçant en particulier) ne pourra résoudre efficacement le problème.

La démarche

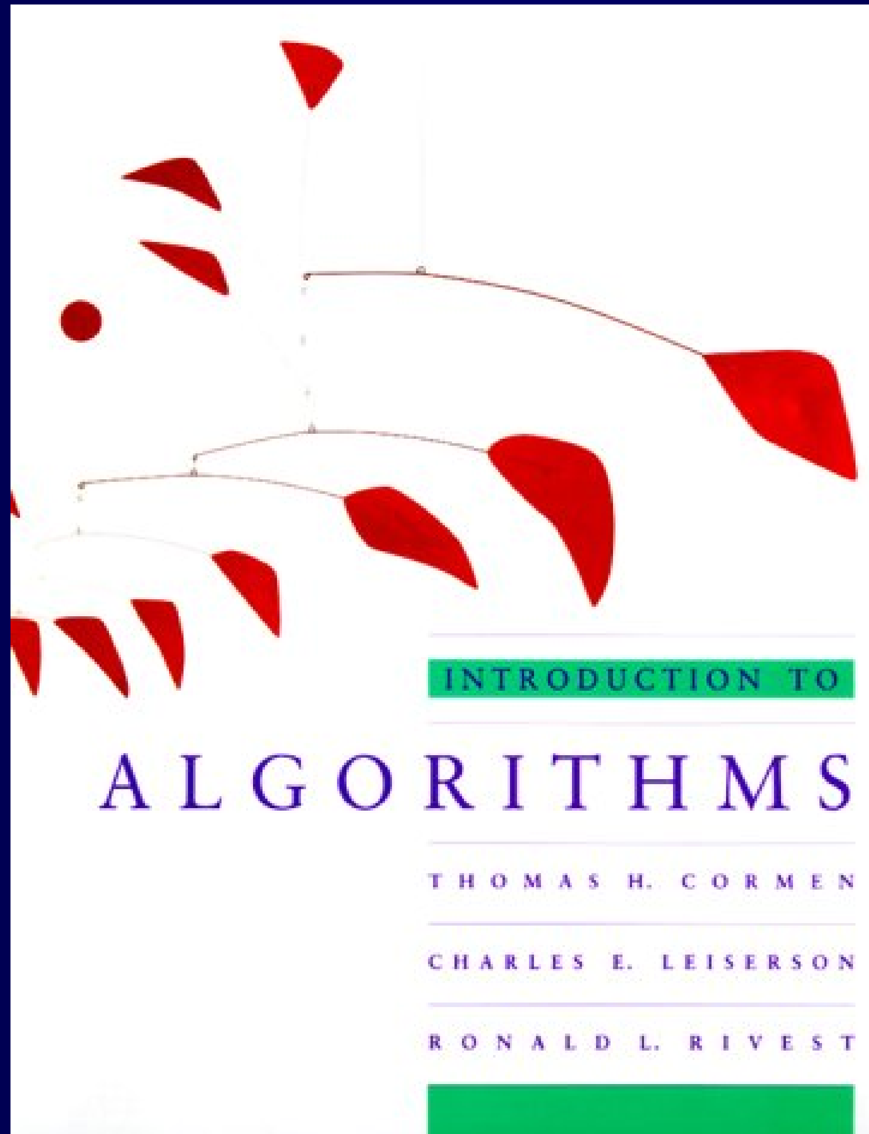
1. Définition d’un modèle de calcul
2. Définition de la complexité dans ce modèle
3. Définition des classes de difficulté
4. Définition d’une classe de problèmes difficiles

Bibliographie

Le LIVRE

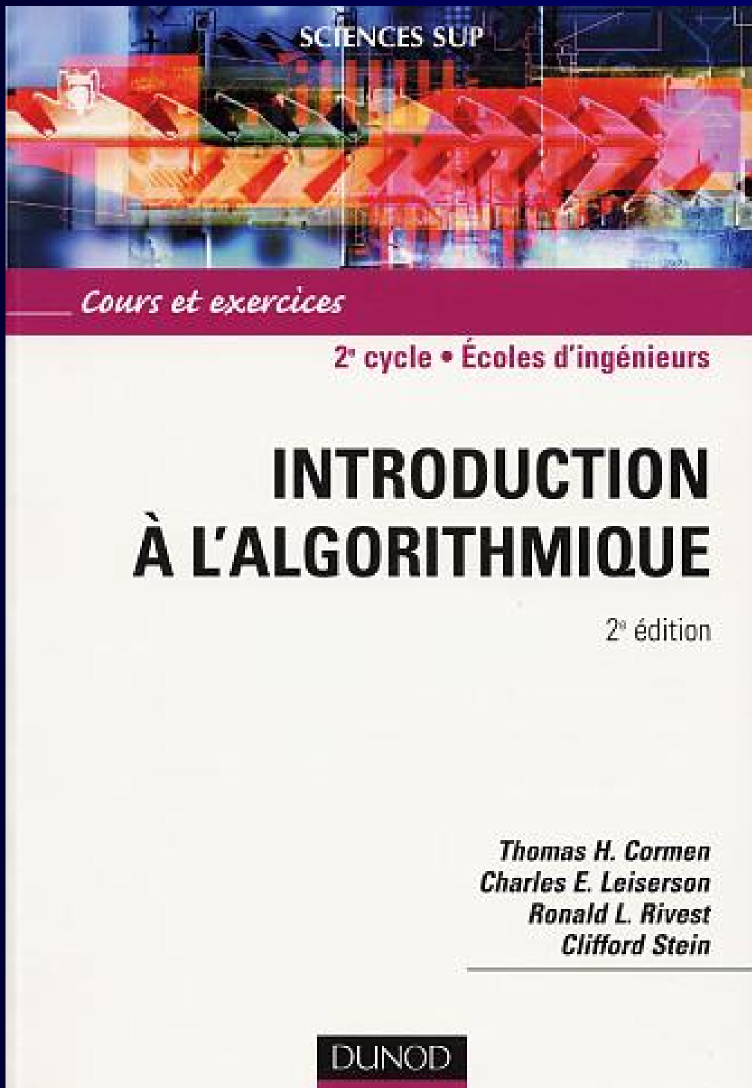
Surtout de l'Algorithmique, mais avec un chapitre consacré aussi à la complexité :

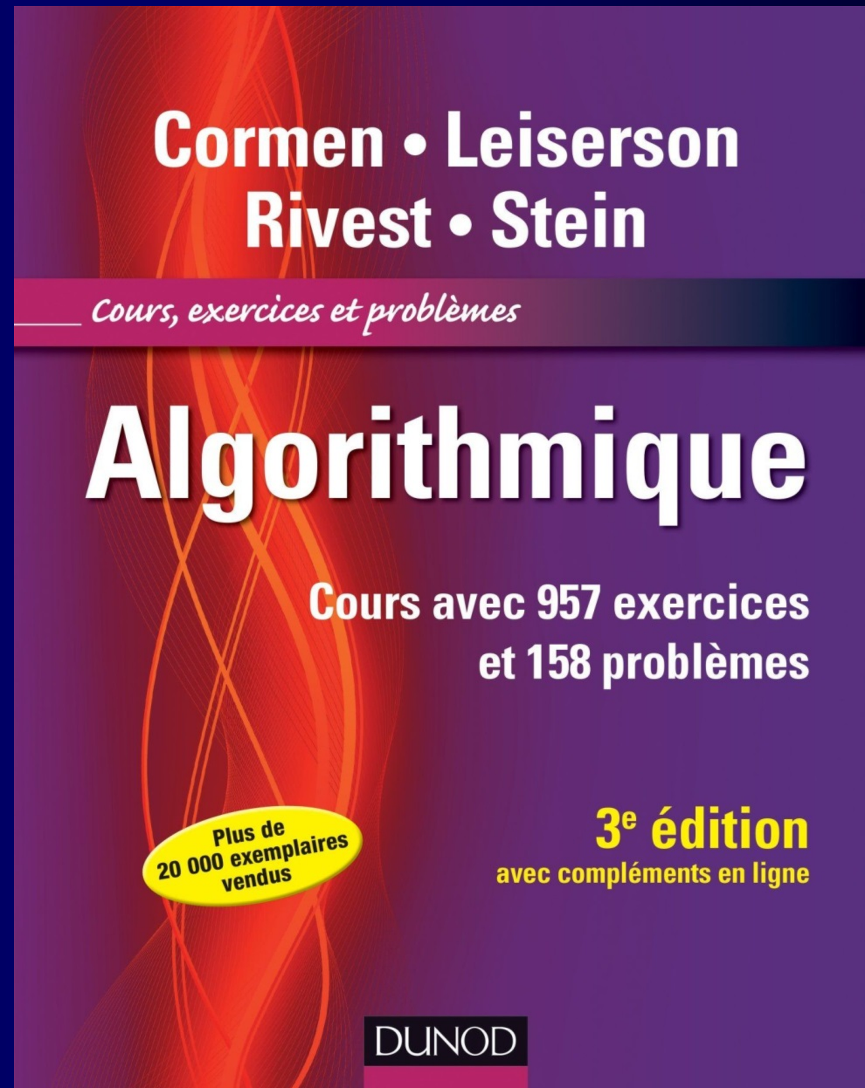
Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest : Introduction to algorithms, MIT Press, 1990.


INTRODUCTION TO
ALGORITHMS
THOMAS H. CORMEN
CHARLES E. LEISERSON
RONALD L. RIVEST

VF

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein : Introduction à l'algorithmique, *troisième édition*, Dunod, 2010.

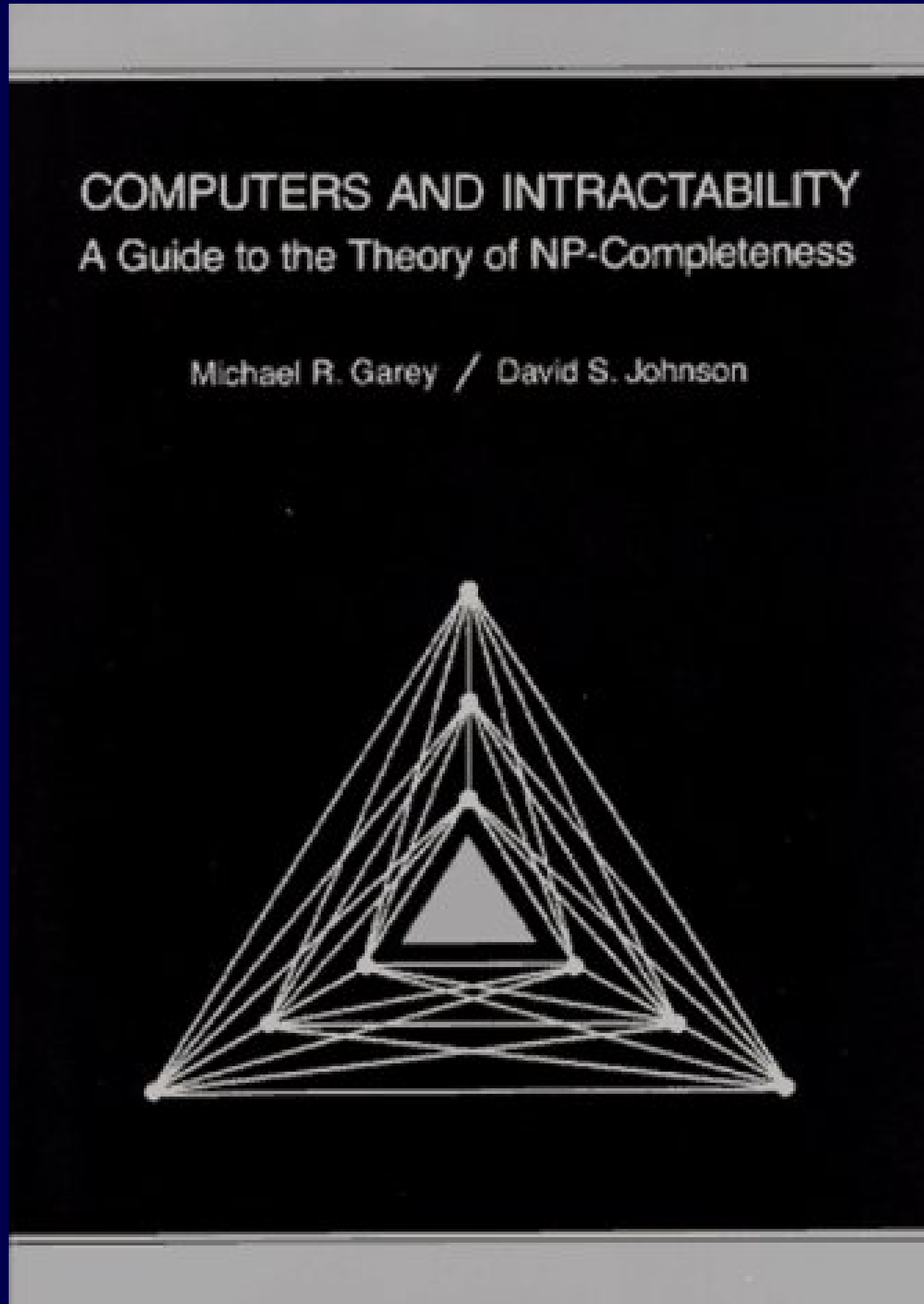

SCIENCES SUP
Cours et exercices
2e cycle • Écoles d'ingénieurs
INTRODUCTION
À L'ALGORITHMIQUE
2e édition
Thomas H. Cormen
Charles E. Leiserson
Ronald L. Rivest
Clifford Stein
DUNOD


Cormen • Leiserson
Rivest • Stein
Cours, exercices et problèmes
Algorithmique
Cours avec 957 exercices
et 158 problèmes
Plus de
20 000 exemplaires
vendus
3e édition
avec compléments en ligne
DUNOD

M.R. Garey et D.S. Johnson :

"*Computers and intractability. A guide to the theory of NP-completeness*",

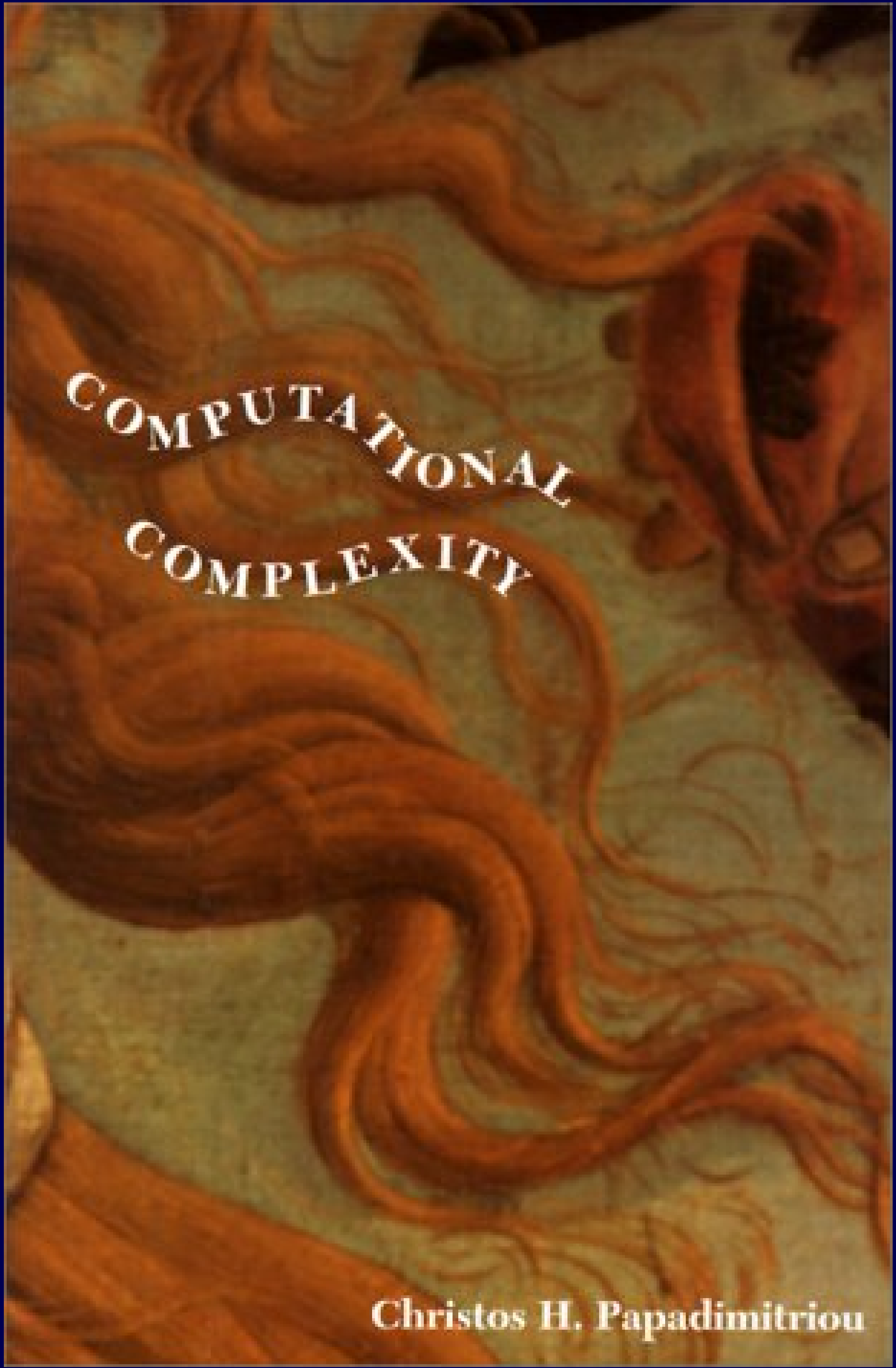
Freeman, New York, 1979.


COMPUTERS AND INTRACTABILITY
A Guide to the Theory of NP-Completeness
Michael R. Garey / David S. Johnson

C.H. Papadimitriou :

"*Computational Complexity*",

Addison-Wesley, 1994.


COMPUTATIONAL
COMPLEXITY
Christos H. Papadimitriou

K. Mehlhorn :

"Data Structures and Algorithms» vol 2 :"Graph Algorithms and NP-completeness",

Springer-Verlag, 1984.

Le chapitre concernant les problèmes NP-complets est téléchargeable à l'adresse :

http://www.mpi-sb.mpg.de/~mehlhorn/ftp/EATCSmonograph/chapter6.ps

EATCS
Monographs on
Theoretical Computer Science

W. Brauer G. Rozenberg A. Salomaa (Ed.)

KURT MEHLHORN

Data Structures and Algorithms 2:

Graph Algorithms and NP-Completeness

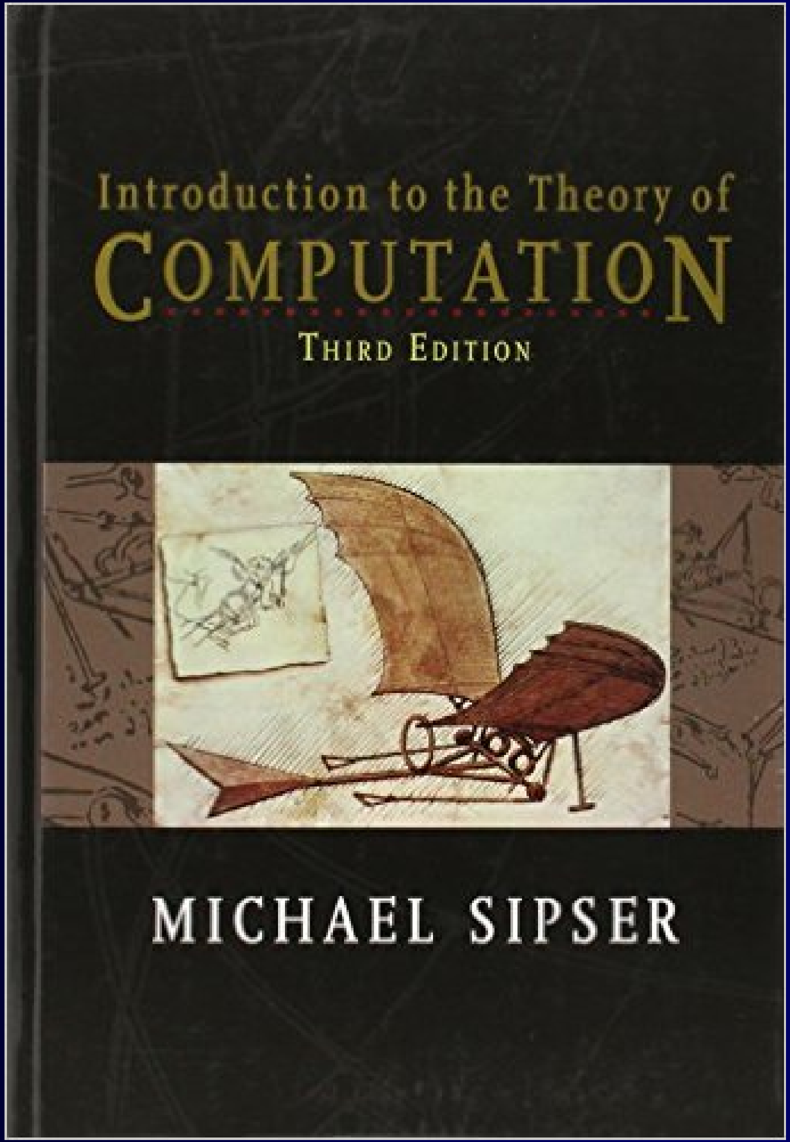
Springer-Verlag
Berlin Heidelberg New York Tokyo

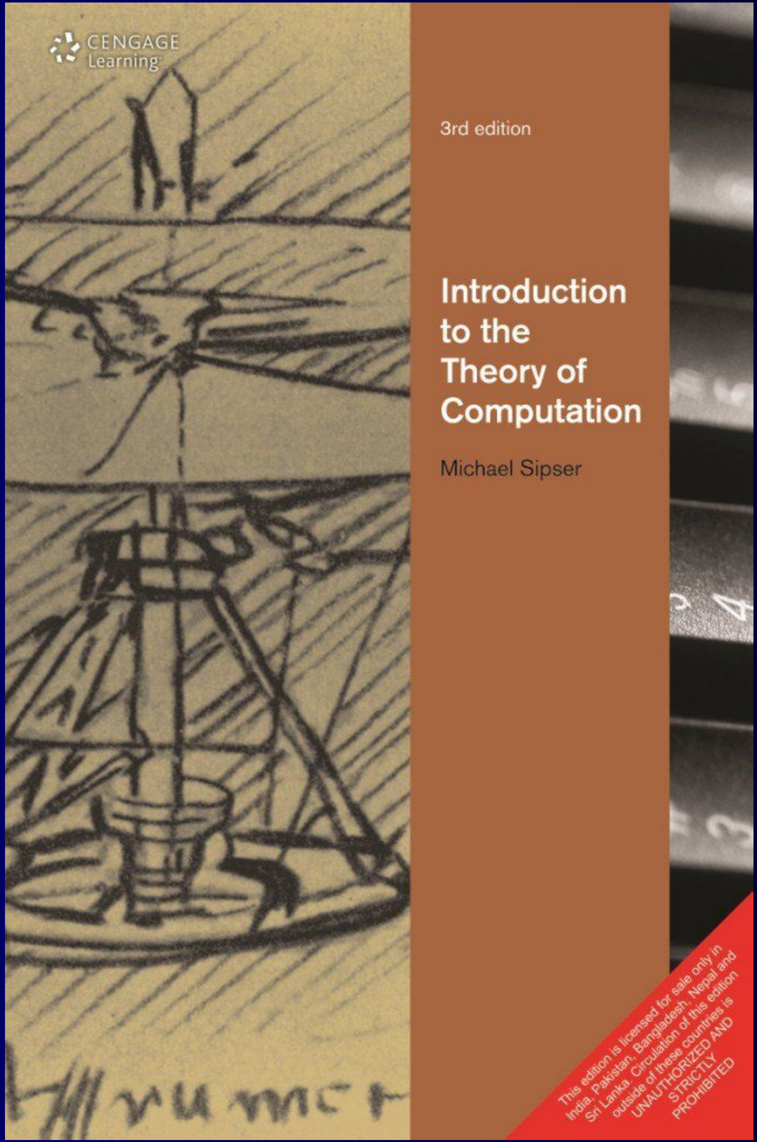
M. Sipser :

"*Introduction to the Theory of Computation*",

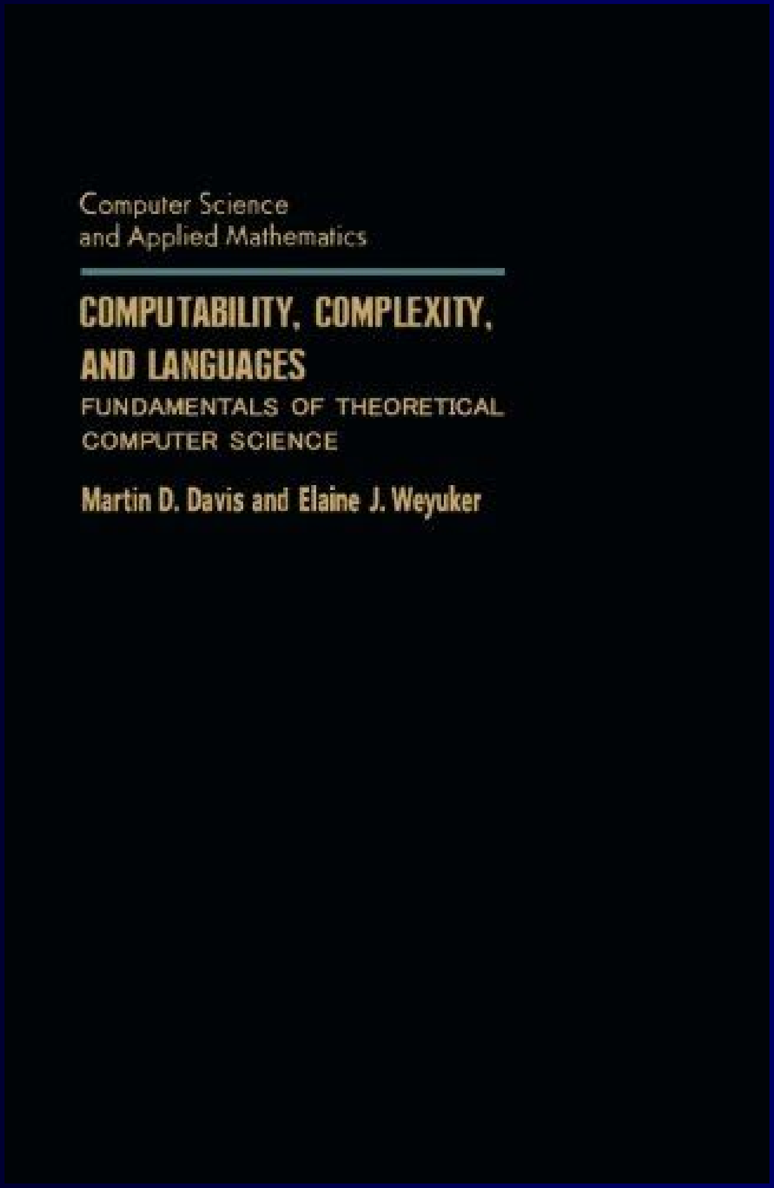
PWS, 1997

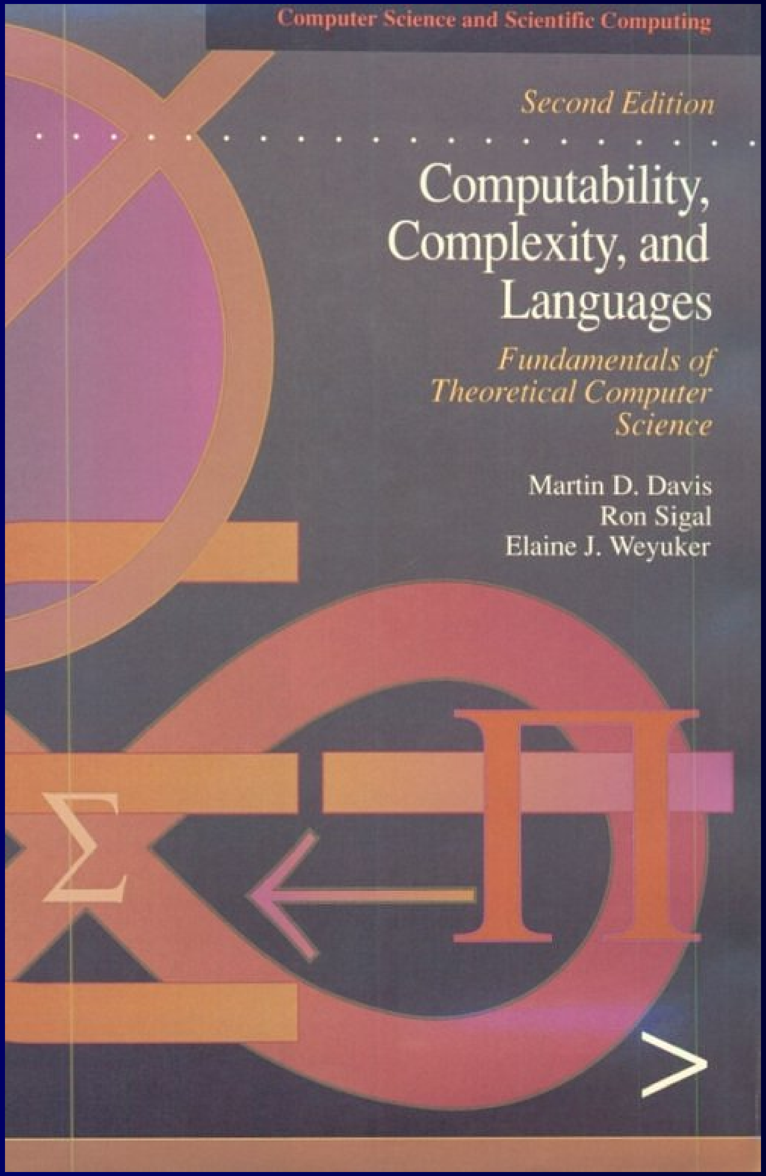
troisième édition 2012


Introduction to the Theory of
COMPUTATION
THIRD EDITION
MICHAEL SIPSER


CENGAGE
Learning
3rd edition
Introduction
to the
Theory of
Computation
Michael Sipser
This edition is licensed for sale only in India, Pakistan, Bangladesh, Nepal and Sri Lanka. Circulation of this edition outside of these countries is UNAUTHORIZED AND STRICTLY PROHIBITED

M.D. Davis, R. Sigal, E.J. Weyuker :
"*Computability, Complexity & Languages*", Academic Press.

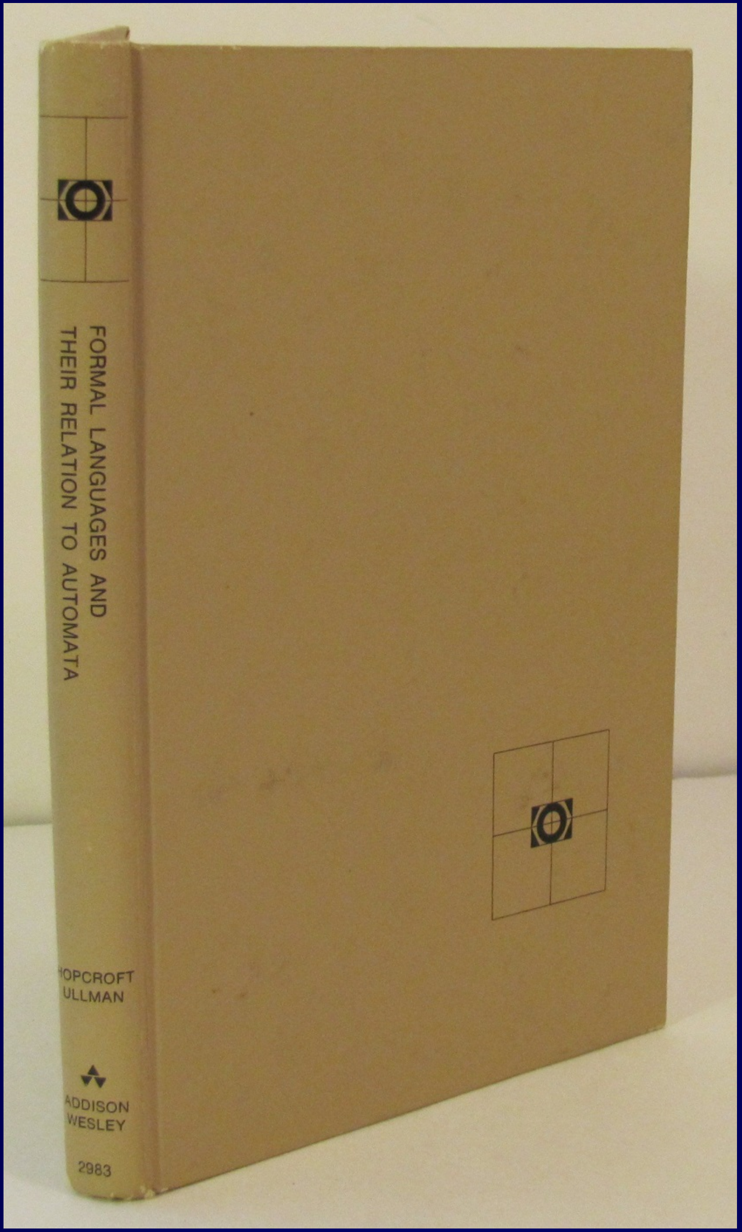

Computer Science
and Applied Mathematics
COMPUTABILITY, COMPLEXITY,
AND LANGUAGES
FUNDAMENTALS OF THEORETICAL
COMPUTER SCIENCE
Martin D. Davis and Elaine J. Weyuker


Computer Science and Scientific Computing
Second Edition
Computability,
Complexity, and
Languages
Fundamentals of
Theoretical Computer
Science
Martin D. Davis
Ron Sigal
Elaine J. Weyuker

J.H. Hopcroft et J.D. Ullman :

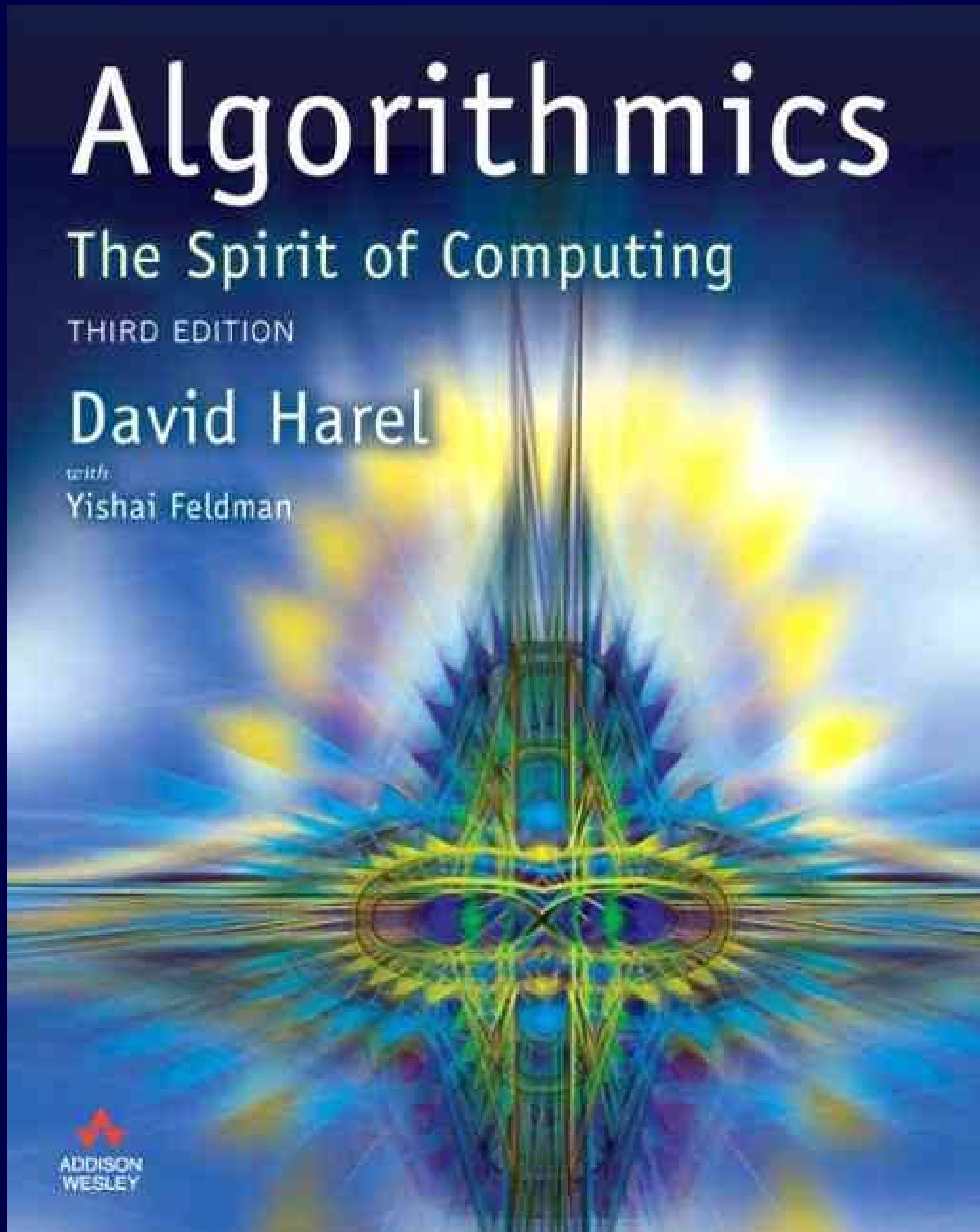
"Formal languages and their relation to automata",

Addison-Wesley, 1969.


FORMAL LANGUAGES AND
THEIR RELATION TO AUTOMATA
HOPCROFT
ULLMAN
ADDISON
WESLEY
2983

D. Harel :

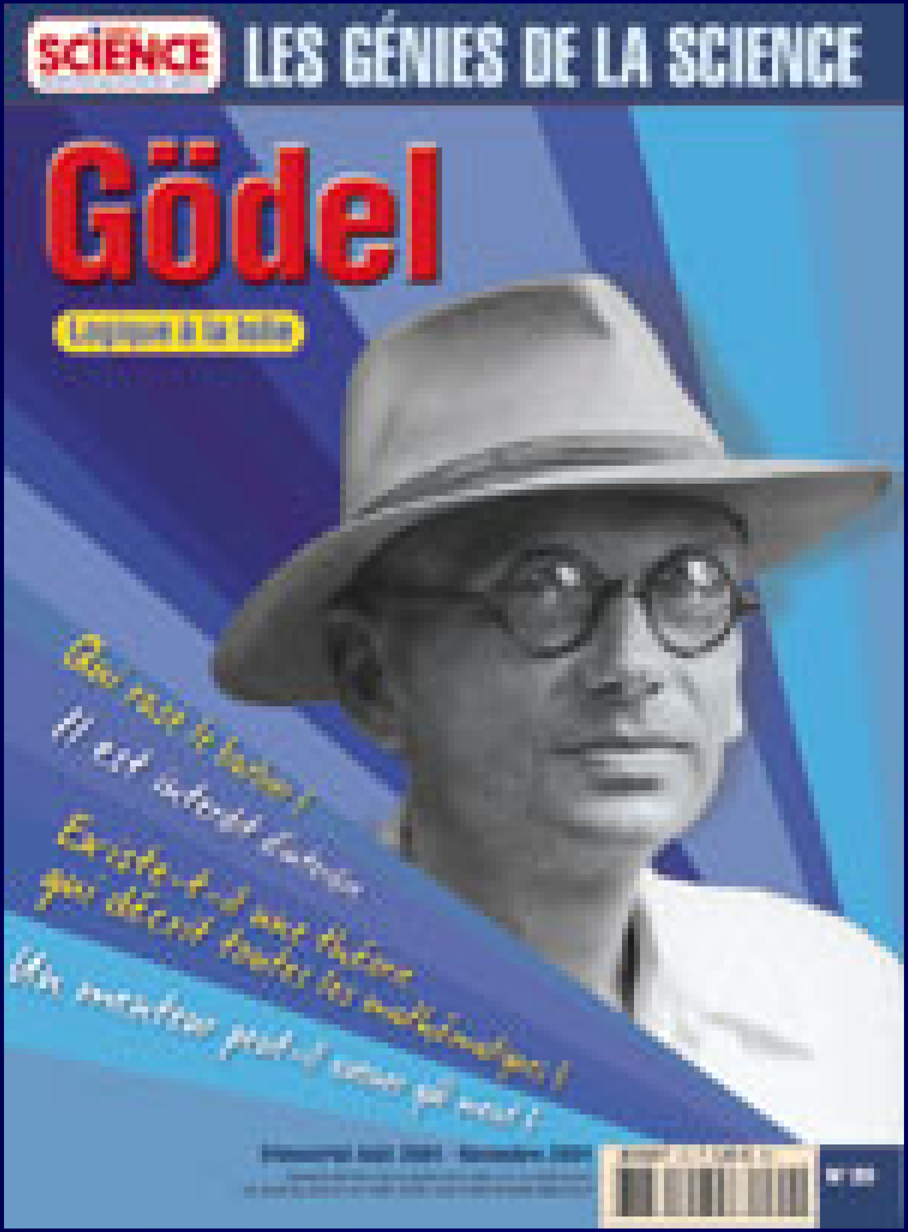
***"Algorithmics : The spirit of computing"*, Addison Wesley.**


Algorithmics
The Spirit of Computing
THIRD EDITION
David Harel
with
Yishai Feldman
ADDISON
WESLEY

Numéro spécial de "Pour la science" consacré à

Kurt Gödel

(série « Les génies de la science », n° 20, 11/2004)


SCIENCE
LES GÉNIES DE LA SCIENCE
Gödel

Un modèle de calcul

La MACHINE de TURING

Les machines de Turing

- Ce modèle de calcul date de 1936 et est dû à Turing.
- Historiquement, c'est le premier modèle *«plutôt informatique»* et sûrement le modèle le plus connu.

Les machines de Turing

Une machine de Turing :

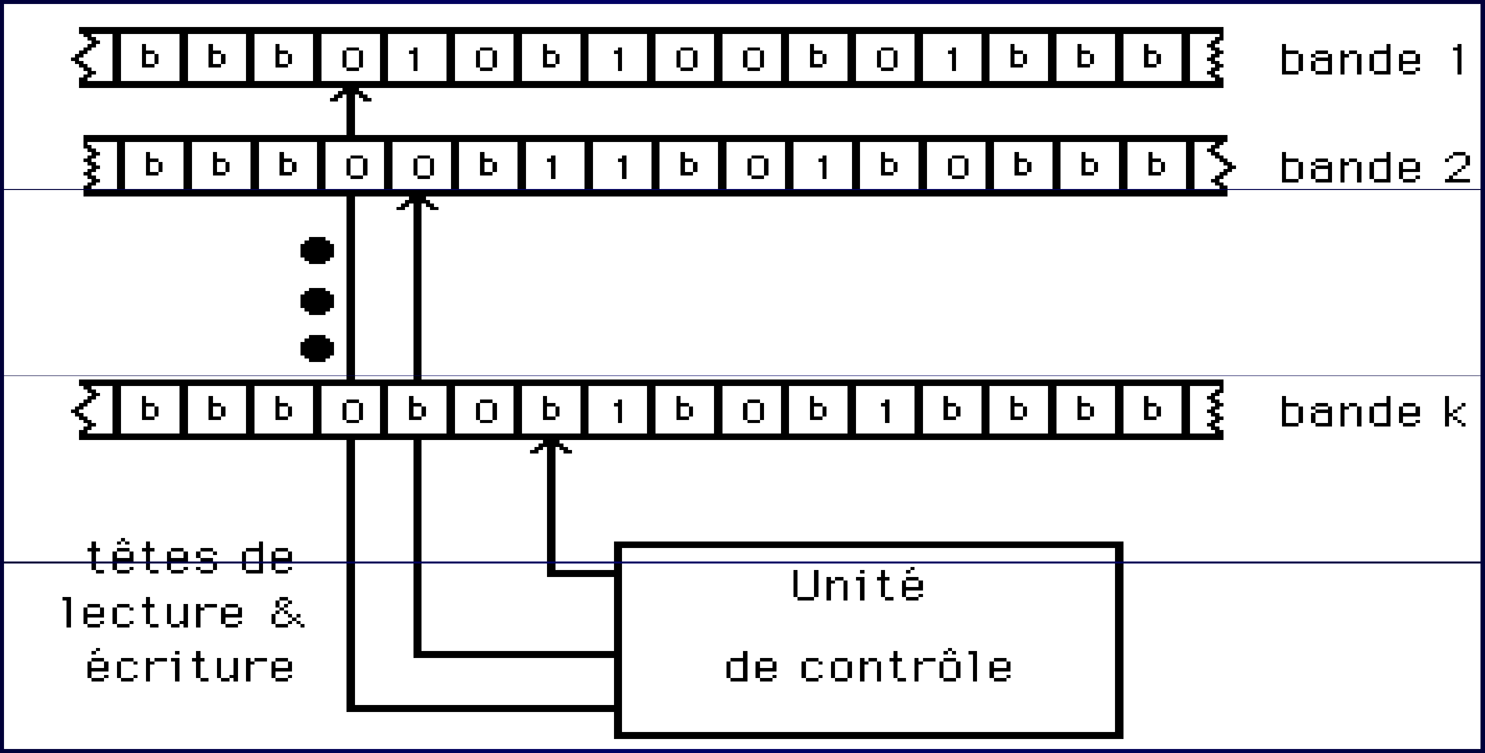
- une ou plusieurs bandes de lecture/écriture

- une unité de contrôle finie (une sorte d'automate fini).

Les machines de Turing

- chaque bande est divisée en cases
- chaque case contient un symbole d'un alphabet fini
- les bandes sont infinies des deux côtés (il y a un nombre infini de cases à gauche et à droite).
- on dispose de têtes de lecture et écriture permettant de lire le contenu d'une case et d'écrire un des symboles de l'alphabet dans cette case.

Les machines de Turing



Les machines de Turing

Définition : une machine de Turing à k bandes est décrite par un septuplet (Q, T, I, d, B, q0, qf). Les composantes de ce septuplet sont :

- Q - un ensemble fini d'états de la machine.
- T - un ensemble fini de symboles qu'on peut utiliser sur les bandes.
- I - un sous-ensemble de l'ensemble des symboles, les symboles de données.
- δ - la fonction de transition : c'est une fonction de $Q \times T^k$ vers $Q \times (T \times \{G,D,S\})^k$.
- B - un symbole, désignant le blanc, appartenant à T\I.
- q_0 - l'état initial.
- q_f - l'état final.

Les machines de Turing

- Un exemple : additionneur binaire

  La machine utilise trois bandes.

  Les deux premières pour les données, la troisième pour le résultat.

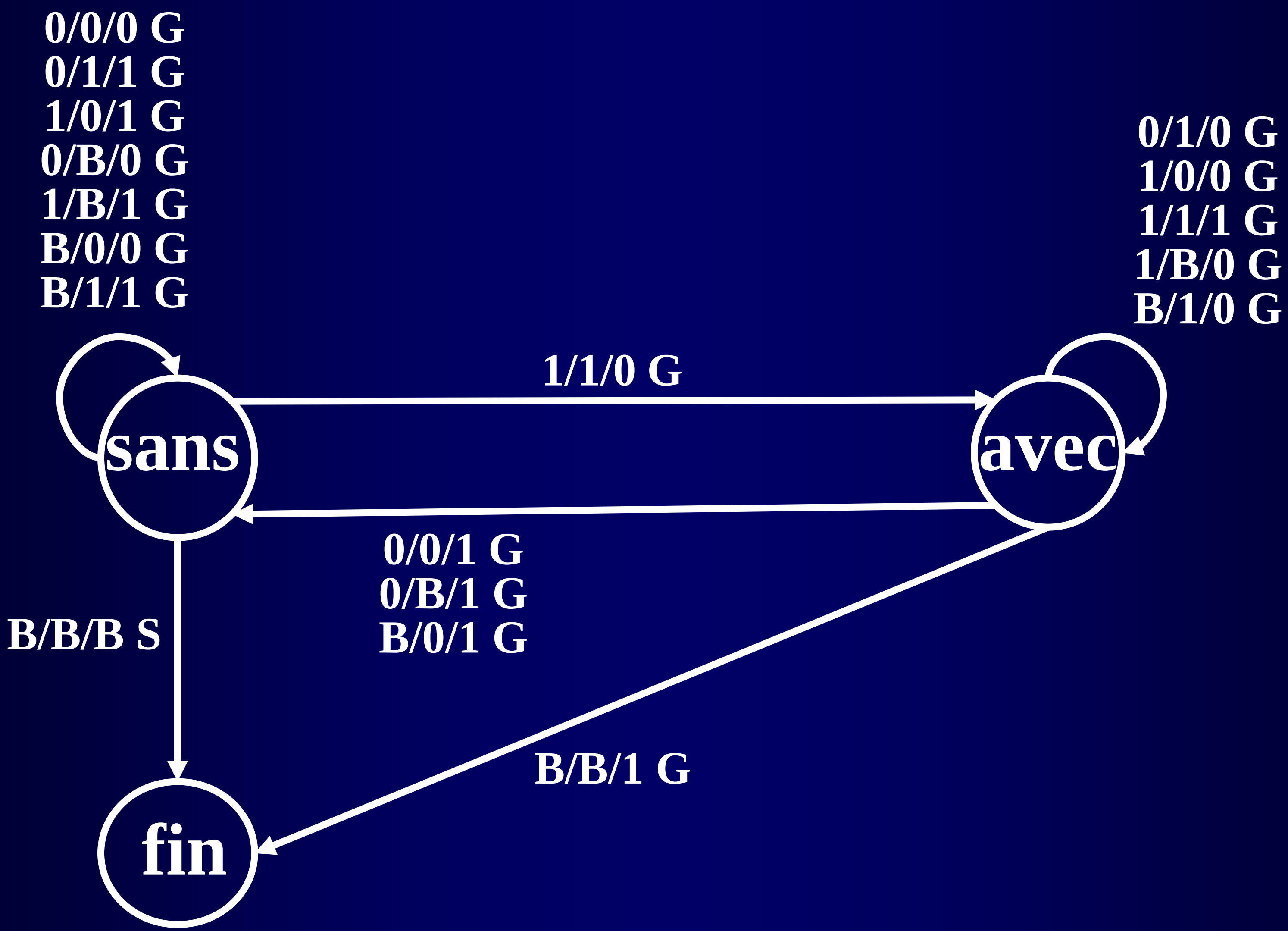
  Les têtes sont positionnés à droite.

  Les déplacement sont toujours à gauche.

  Dans la table de transition nous omettons tout ce qui n’est pas absolument nécessaire …

Etat	Lect1	Lect2	Ecr3	Depl3	N. état	Rem
q_0	0	0	0	G	q_0	sans
	0	1	1	G	q_0	
	1	0	1	G	q_0	
	1	1	0	G	q_1	
	0	B	0	G	q_0	
	1	B	1	G	q_0	
	B	0	0	G	q_0	
	B	1	1	G	q_0	

Etat	Lect1	Lect2	Ecr3	Depl3	N. état	Rem
q_1	0	0	1	G	q_0	avec
	0	1	0	G	q_1	
	1	0	0	G	q_1	
	1	1	1	G	q_1	
	0	B	1	G	q_0	
	1	B	0	G	q_1	
	B	0	1	G	q_0	
	B	1	0	G	q_1	


0/0/0 G
0/1/1 G
1/0/1 G
0/B/0 G
1/B/1 G
B/0/0 G
B/1/1 G
0/1/0 G
1/0/0 G
1/1/1 G
1/B/0 G
B/1/0 G
1/1/0 G
sans
avec
0/0/1 G
0/B/1 G
B/0/1 G
B/B/B S
B/B/1 G
fin

Variantes de la machine

Les différents modèles se distinguent surtout selon les trois critères suivants :

- le nombre de bandes
  - une bande
  - deux bandes
  - …
- la nature des bandes
  - infinis des deux cotés
  - infinis d’un seul coté
- le déterminisme ou le non-déterminisme de la machine.

Variantes de la machine

Il existent d'autres variantes aussi, comme par exemple celle où on admet une seule opération à la fois

- changement d'état

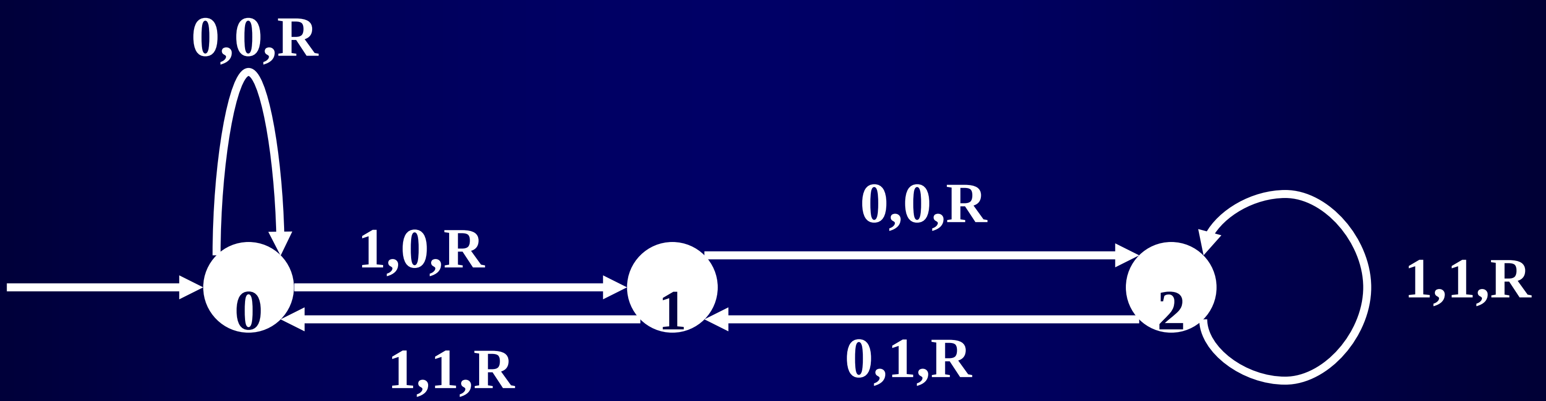
ou

- déplacement

ou

- écriture

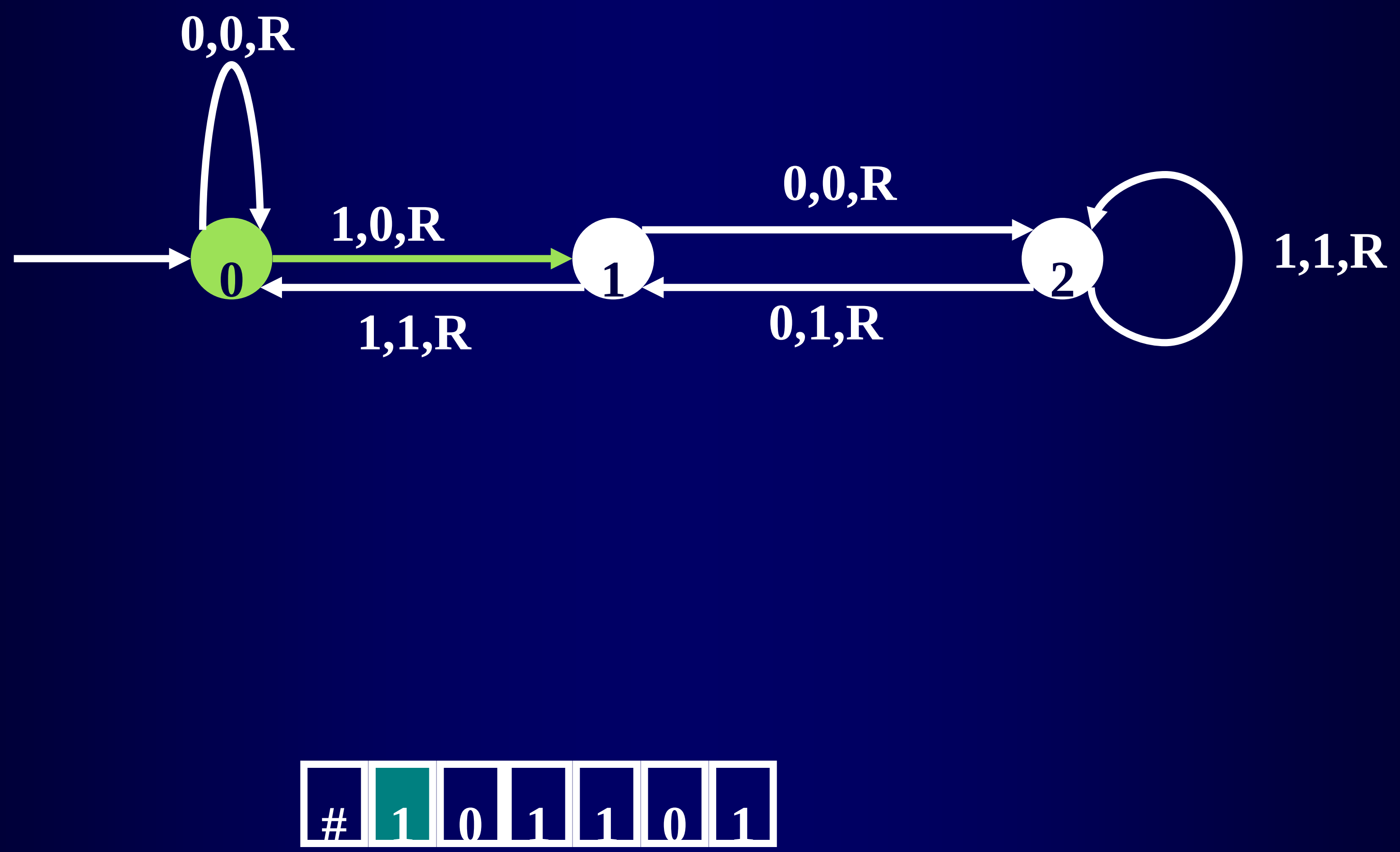
Exemple : division par 3



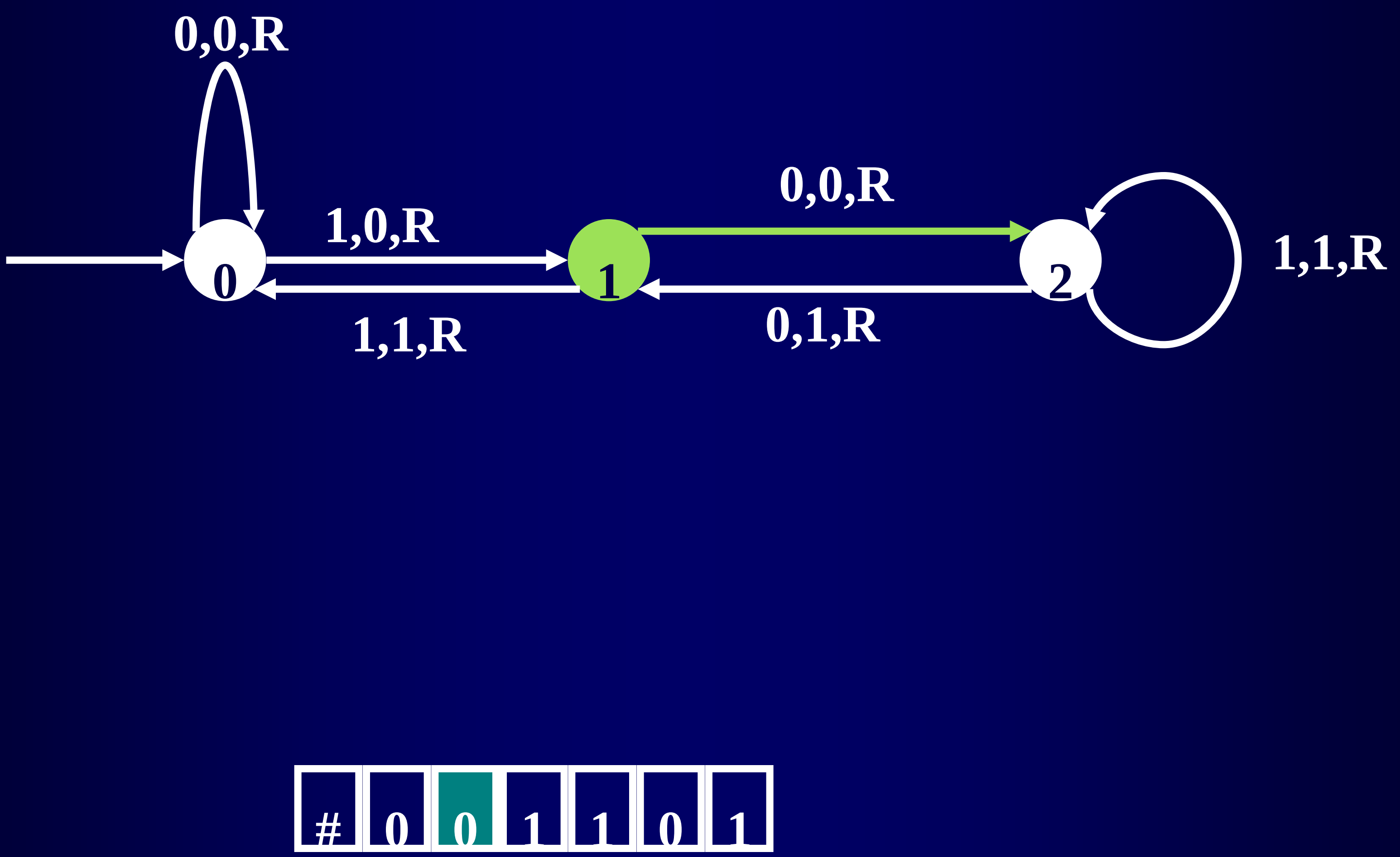
45:3=

#	1	0	1	1	0	1

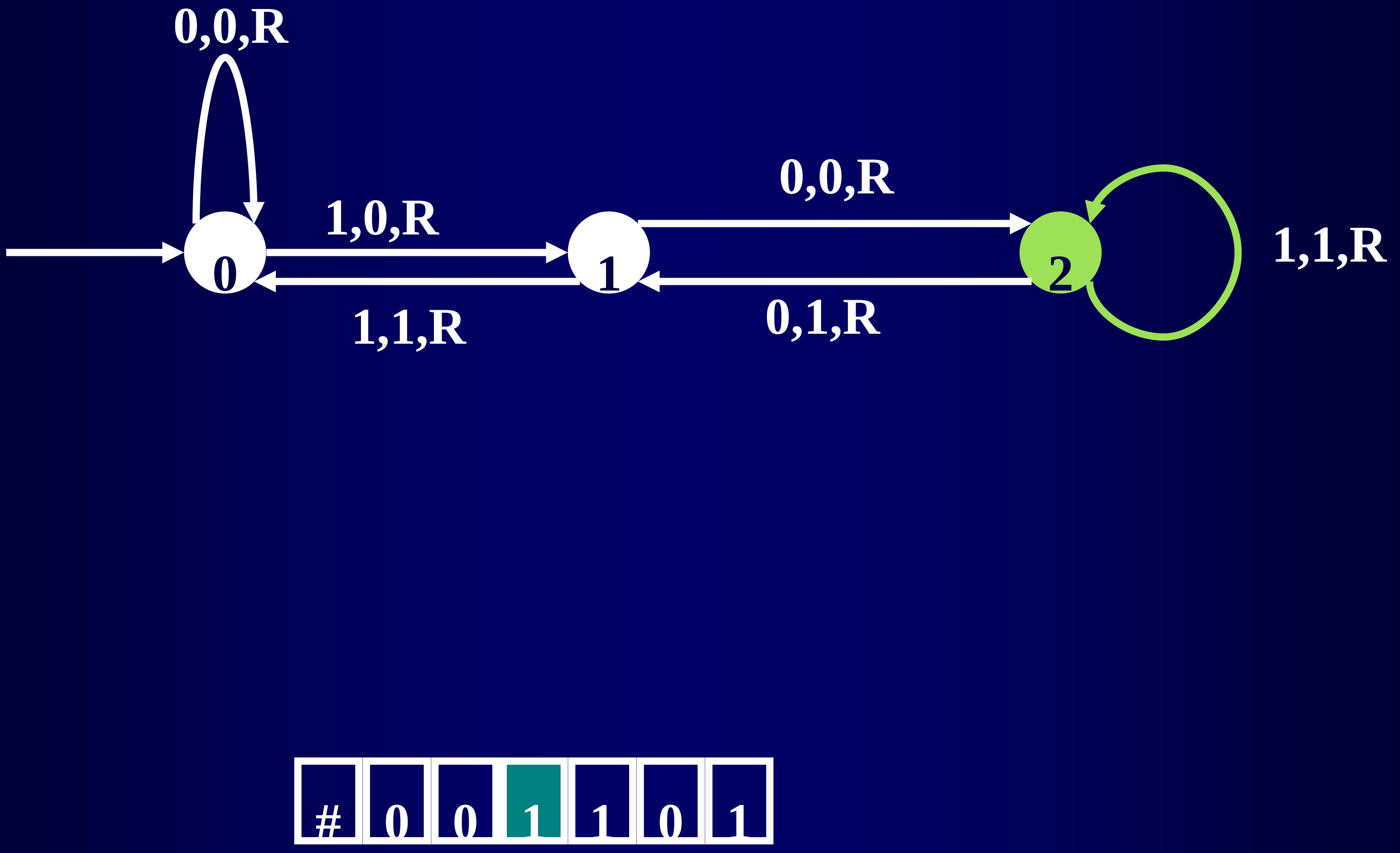
Exemple : division par 3



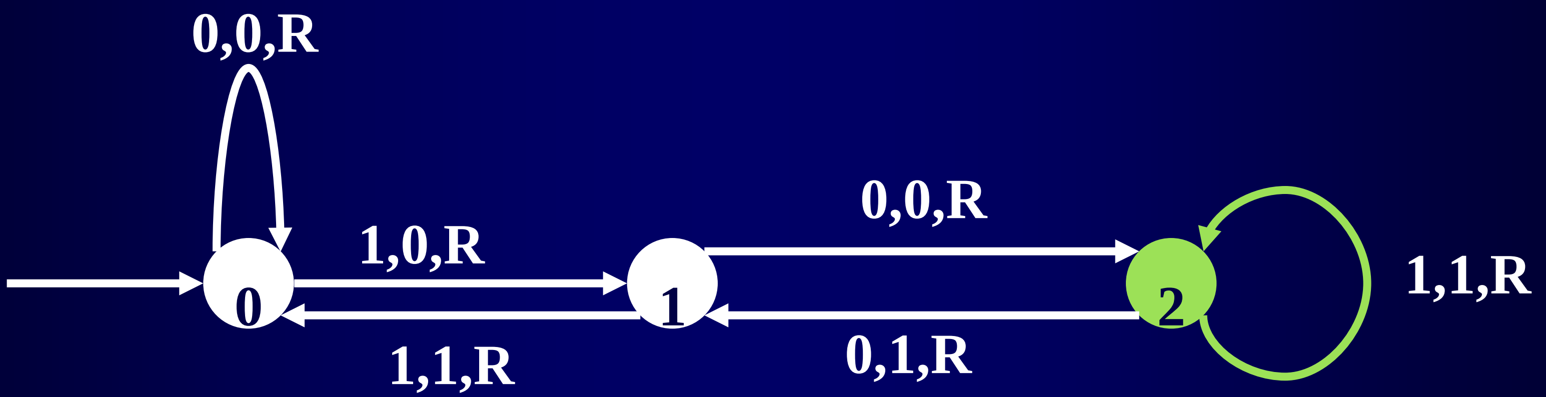
Exemple : division par 3

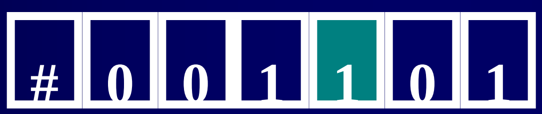


Exemple : division par 3

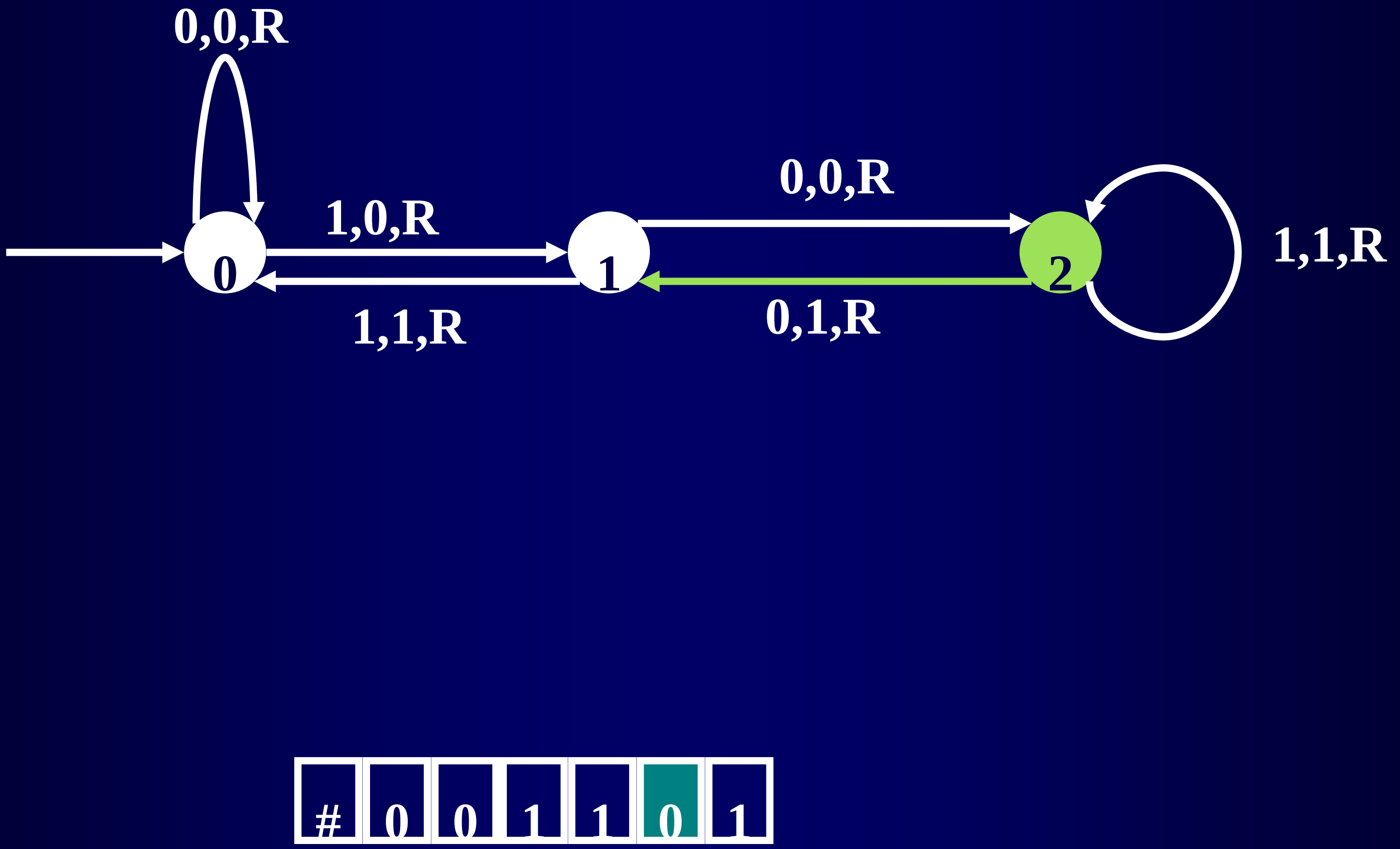


Exemple : division par 3

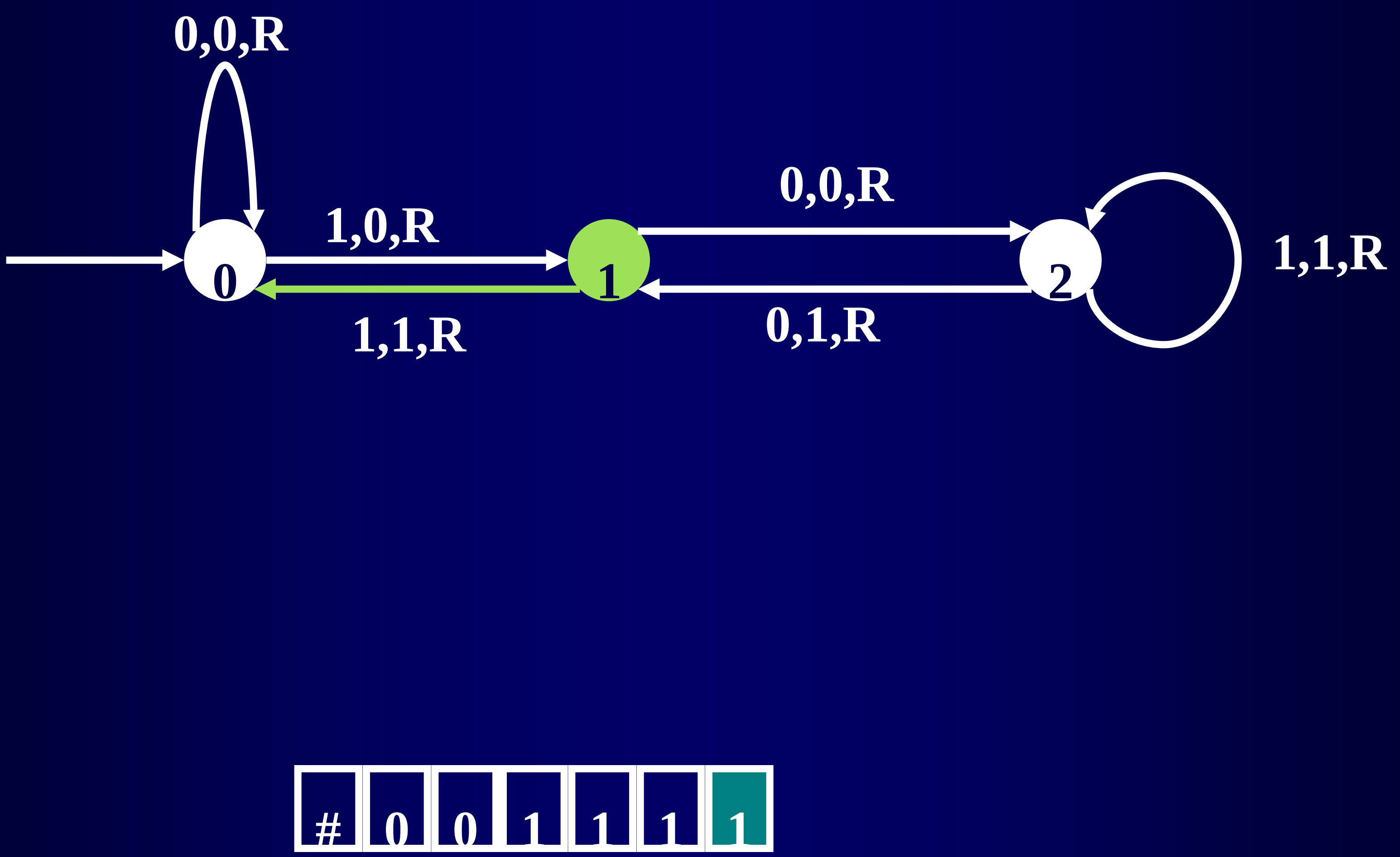




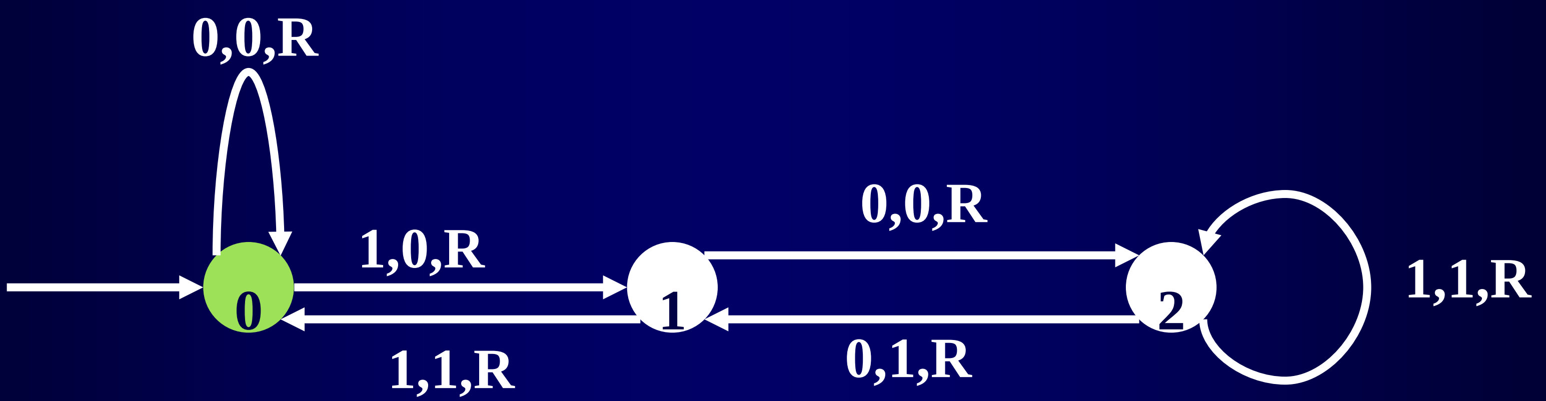
Exemple : division par 3



Exemple : division par 3



Exemple : division par 3



45:3=15