

BROUILLON - Quelques machines de Turing

Christophe BAL

7 Février 2020

*Document, avec son source L^AT_EX, disponible sur la page
<https://github.com/bc-writing/drafts>.*

Mentions « légales »

Ce document est mis à disposition selon les termes de la licence Creative Commons “Attribution - Pas d’utilisation commerciale - Partage dans les mêmes conditions 4.0 International”.



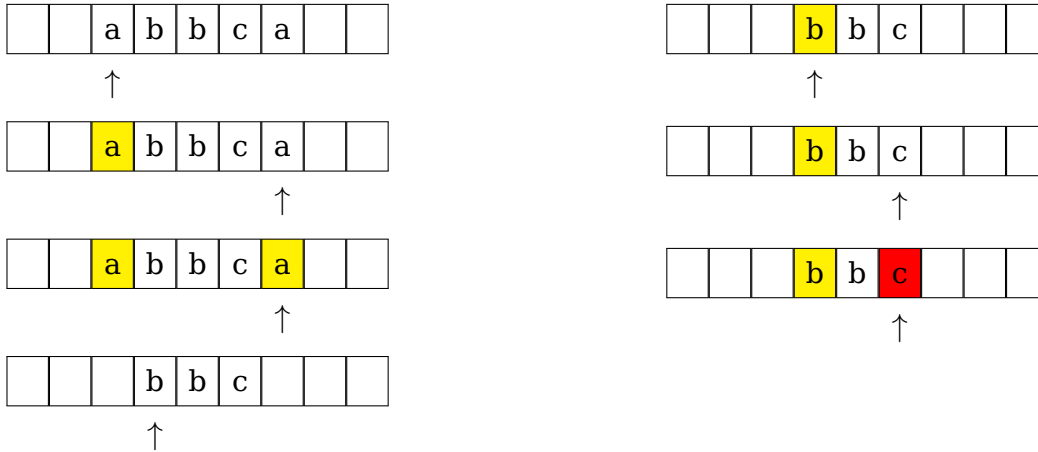
Table des matières

1 Détecter les palindromes	2
1.1 Un exemple avec le mot abbca	2
1.2 Un exemple avec le mot abbcbbba	2
1.3 La table de transition	3
1.4 Coder pour tester	4

1 Détecter les palindromes

1.1 Un exemple avec le mot abbca

Voici les grandes étapes présentées sur deux colonnes.



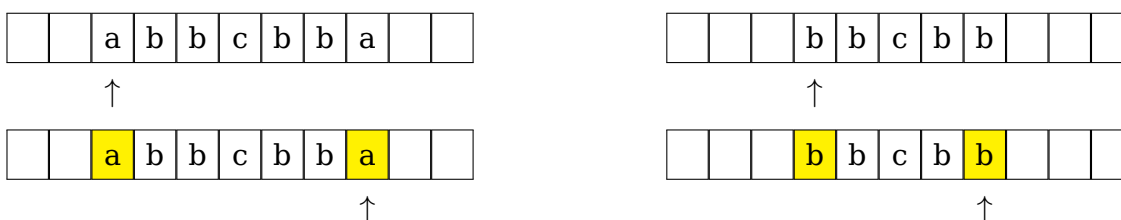
Qu'a-t-on fait ?

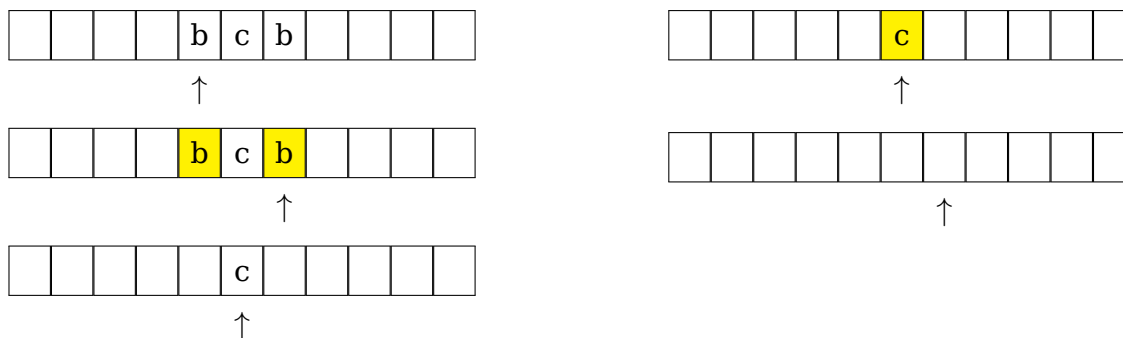
1. On note la lettre pointée par la tête de lecture symbolisée ici par la flèche verticale ↑. Commence alors une phase de recherche d'une lettre connue.
2. On avance tant que l'on ne rencontre pas une case vide. Une fois celle-ci repérée on revient d'une case en arrière.
3. On compare alors la lettre pointée par la tête de lecture avec celle de la phase de recherche en cours. Deux cas sont possibles.
 - a) Si les lettres sont différentes alors on ne fait plus rien. On a un état bloquant et le mot n'est pas validé.
 - b) Si les lettres sont identiques alors on efface la lettre en cours puis on va vers la gauche jusqu'à la prochaine case vide. Une fois celle-ci trouvée, on avance d'une case vers la droite pour effacer son contenu, puis on avance d'une autre case vers la droite pour recommencer les actions à partir du point 1.

La méthode ci-dessus est en fait incomplète comme nous allons le voir dans la section suivante avec un exemple de palindrome à repérer.

1.2 Un exemple avec le mot abbcbbba

On reprend la même méthodologie que dans la section précédente en la complétant sur la fin. Voici les étapes principales.





Quelle est la nouveauté ici ? Il faut juste ajouter après chaque nouvelle phase suite à un effacement la règle suivante : si la tête de lecture est sur une case vide alors on est dans un état final qui marque la validation d'un mot palindromique.

1.3 La table de transition

Formalisons les étapes utilisées dans les exemples précédents pour des mots sur l'alphabet $\{a; b; c\}$ (la méthode est généralisable sans souci). Nous avons besoin des états suivants.

- q_0 est l'état initial.
- q_1 est l'état pour chaque début de nouvelle recherche de lettres identiques en début et en fin de mot à partir de la deuxième lettre.
- Il nous faut des états pour rechercher une lettre identique à droite, ce sera les états d_a , d_b et d_c .
- Les états précédents vont en fait nous permettre d'arriver sur la case vide à droite de la dernière lettre. Nous ajoutons donc les états v_a , v_b et v_c pour valider ou non la dernière lettre.
- Il nous faut aussi un état g pour retourner vers la gauche recommencer l'analyse avec la lettre suivante ainsi qu'un état e pour effacer la 1^{re} lettre qui avait été repérée.
- Enfin l'état f sera notre état final indiquant qu'un mot est bien un palindrome.

Voici finalement la table des transitions de notre machine de Turing pour repérer les palindromes sur l'alphabet $\{a; b; c\}$.

δ	a	b	c	B
q_0	(d_a, a, D)	(d_b, b, D)	(d_c, c, D)	
q_1	(d_a, a, D)	(d_b, b, D)	(d_c, c, D)	(f, B, D)
d_a	(d_a, a, D)	(d_a, b, D)	(d_a, c, D)	(v_a, B, G)
d_b	(d_b, a, D)	(d_b, b, D)	(d_b, c, D)	(v_b, B, G)
d_c	(d_c, a, D)	(d_c, b, D)	(d_c, c, D)	(v_c, B, G)
v_a	(g, B, G)			
v_b		(g, B, G)		
v_c			(g, B, G)	
g	(g, a, G)	(g, b, G)	(g, c, G)	(e, B, D)
e	(q_1, B, D)	(q_1, B, D)	(q_1, B, D)	(q_1, B, D)

1.4 Coder pour tester

Sur le site de téléchargement de ce document, dans le sous-dossier `turing/palindrome`, se trouve le fichier `palindrome.txt` contenant un code utilisable pour des tests manuels sur le site <https://turingmachinesimulator.com>.