Recherche textuelle

0- Introduction

Pourquoi parler de recherche textuelle ?

Qu'est-ce qu'un texte ?

Quelques exemples

- `10101010001111010101010`
- `ATCATAGCAGCAAGGACTACGAT`
- `un texte`
- la concaténation de toutes les pages web

Qu'est-ce qu'un texte ?

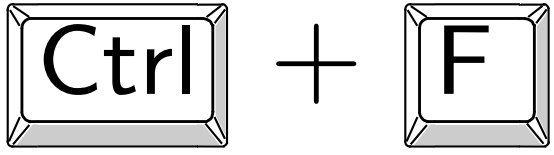
Quelques exemples

- `10101010001111010101010`
- `ATCATAGCAGCAAGGACTACGAT`
- `un texte`
- la concaténation de toutes les pages web

Définition

Un texte est une suite finie de symboles

Quelle recherche textuelle ?



Recherche dans un document

Sur un moteur de recherche en ligne

Recherche dans un texte connu à l'avance (livres, sites...)

Index

C

Quand ne construit-on pas d'index ?

Lorsque le texte possède certaines propriétés

Quand ne construit-on pas d'index ?

Lorsque le texte possède certaines propriétés

- court

Quand ne construit-on pas d'index ?

Lorsque le texte possède certaines propriétés

- court
- modifiable

Quand ne construit-on pas d'index ?

Lorsque le texte possède certaines propriétés

- court
- modifiable
- non connu à l'avance

Notez qu'un PDF de 1 000 pages ne remplit aucun de ces critères

Quand ne construit-on pas d'index ?

Lorsque le texte possède certaines propriétés

- court
- modifiable
- non connu à l'avance

Notez qu'un PDF de 1 000 pages ne remplit aucun de ces critères

Comment rechercher dans un texte, sans index ?

1- Recherche naïve

Parcours de texte – recherche naïve

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

Puis-je trouver *le mot* $P =$ atatac ?

Parcours de texte – recherche naïve

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

✓ ✓ ✓ ✗

a t a t a c

Puis-je trouver *le mot* $P =$ atatac ?

Parcours de texte – recherche naïve

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

a t a t a c ✓ ✓ ✓ ✗

a t a t a c ✗

Puis-je trouver *le mot* $P =$ atatac ?

Parcours de texte – recherche naïve

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

a t a t a c

a t a t a c

a t a t a c

Puis-je trouver *le mot* $P =$ atatac ?

Parcours de texte – recherche naïve

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

a t a t a c

a t a t a c

a t a t a c

a t a t a c

Puis-je trouver *le mot* $P =$ atatac ?

Parcours de texte – recherche naïve

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

a t a t a c

a t a t a c

a t a t a c

a t a t a c

Puis-je trouver *le mot* $P =$ atatac ?

Tester la présence de P à chaque position de T

2- L'algorithme du bon caractère

Optimiser le parcours du texte – recherche du bon caractère

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

a t a t a c ✓ ✓ ✓ ✗

a t a t a c ✗

Optimiser le parcours du texte – recherche du bon caractère

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

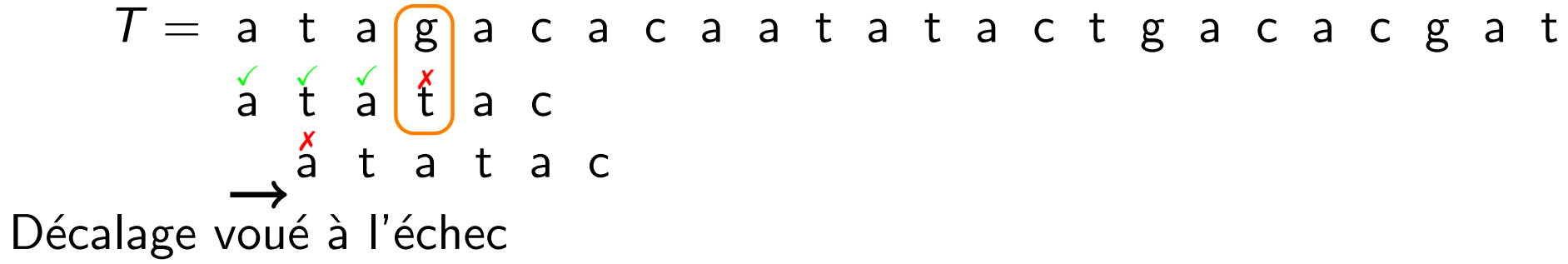
a t a t a c

a t a t a c

$\longrightarrow$

Décalage voué à l'échec

Optimiser le parcours du texte – recherche du bon caractère



Décalage voué à l'échec

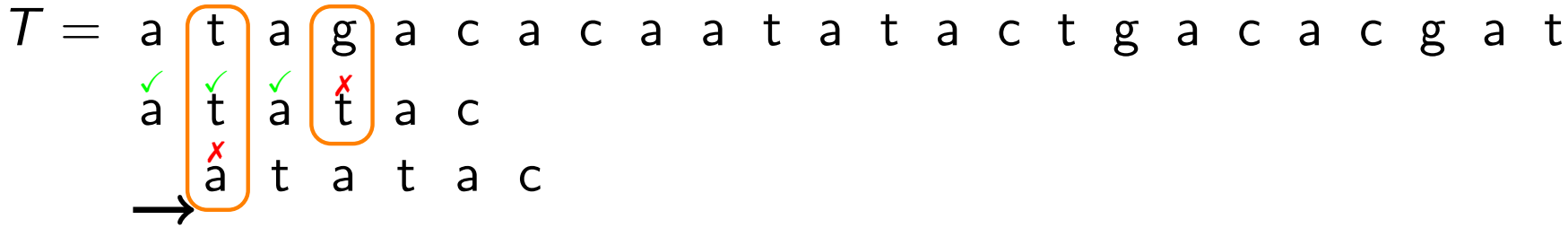
1. La première tentative a échoué en comparant un g. Il n'y en a pas dans P.

Optimiser le parcours du texte – recherche du bon caractère

Décalage voué à l'échec

1. La première tentative a échoué en comparant un g. Il n'y en a pas dans P.
2. En décalant de 1 on met un a en face d'un t.

Optimiser le parcours du texte – recherche du bon caractère



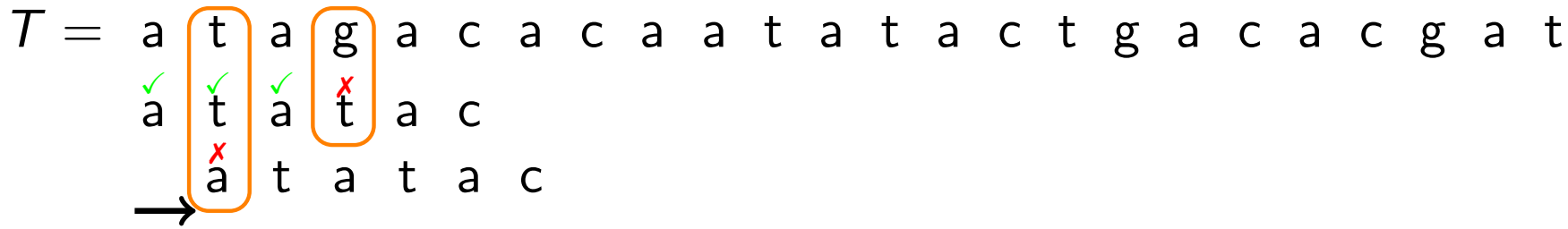
Décalage voué à l'échec

1. La première tentative a échoué en comparant un g. Il n'y en a pas dans P.
2. En décalant de 1 on met un a en face d'un t.

Pour chaque symbole, enregistrons sa dernière position dans $P[0 \cdots i]$, avec

$P =$ a t a t a c (positions 0 1 2 3 4 5)

Optimiser le parcours du texte – recherche du bon caractère



Décalage voué à l'échec

1. La première tentative a échoué en comparant un g. Il n'y en a pas dans P.
2. En décalant de 1 on met un a en face d'un t.

Pour chaque symbole, enregistrons sa dernière position dans $P[0 \cdots i]$, avec

$P =$ a t a t a c (indices 0 1 2 3 4 5)

i	a	c	g	t
0	0	-	-	-
1	0	-	-	1
2	2	-	-	1
3	2	-	-	3
4	4	-	-	3
5	4	5	-	3

Parcours du texte avec la recherche du bon caractère

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

a t a t a c ✓ ✓ ✓ ✗

i	a	c	g	t
0	0	-	-	-
1	0	-	-	1
2	2	-	-	1
3	2	-	-	3
4	4	-	-	3
5	4	5	-	3

Parcours du texte avec la recherche du bon caractère

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

a t a t a c

a t a t a c

i	a	c	g	t
0	0	-	-	-
1	0	-	-	1
2	2	-	-	1
3	2	-	-	3
4	4	-	-	3
5	4	5	-	3

Parcours du texte avec la recherche du bon caractère

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

a t a t a c ✓ ✓ ✓ ✗

a t a t a c ✓ ✗

a t a t a c ✓ ✗

i	a	c	g	t
0	0	-	-	-
1	0	-	-	1
2	2	-	-	1
3	2	-	-	3
4	4	-	-	3
5	4	5	-	3

Parcours du texte avec la recherche du bon caractère

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

a t a t a c ✓ ✓ ✓ ✗

a t a t a c ✓ ✗

a t a t a c ✓ ✗

a t a t a c ✓ ✗

i	a	c	g	t
0	0	-	-	-
1	0	-	-	1
2	2	-	-	1
3	2	-	-	3
4	4	-	-	3
5	4	5	-	3

Parcours du texte avec la recherche du bon caractère

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

a t a t a c

a t a t a c

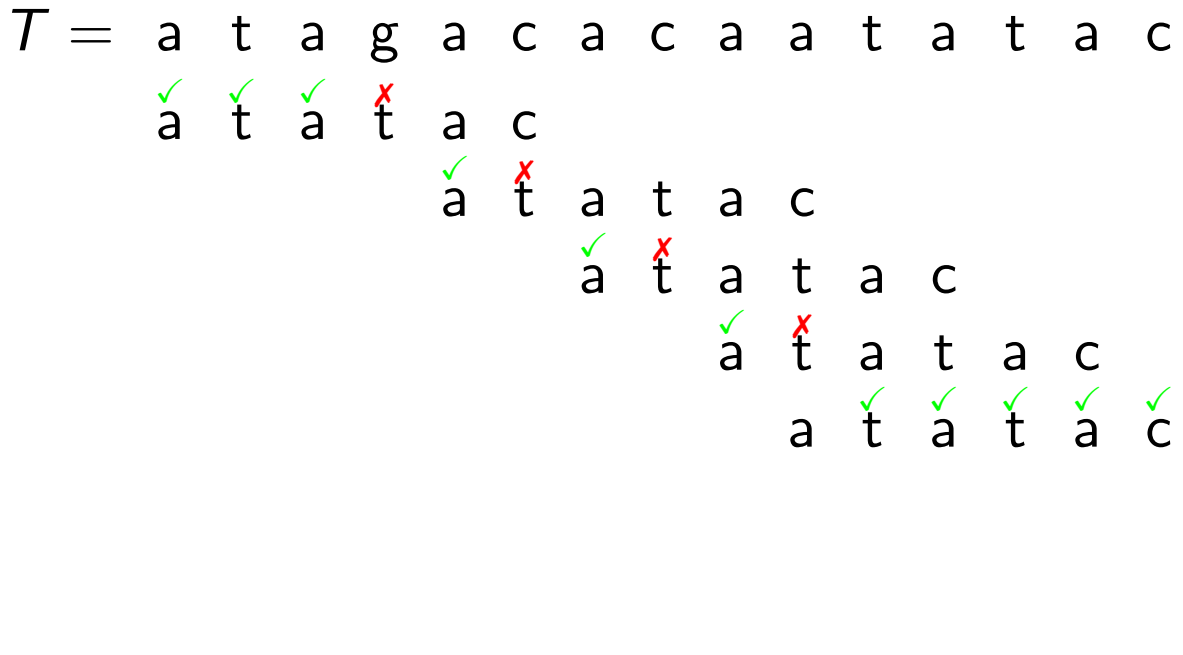
a t a t a c

a t a t a c

a t a t a c

i	a	c	g	t
0	0	-	-	-
1	0	-	-	1
2	2	-	-	1
3	2	-	-	3
4	4	-	-	3
5	4	5	-	3

Parcours du texte avec la recherche du bon caractère



i	a	c	g	t
0	0	-	-	-
1	0	-	-	1
2	2	-	-	1
3	2	-	-	3
4	4	-	-	3
5	4	5	-	3

Une lettre du texte n'est jamais comparée plusieurs fois

Parcours du texte avec la recherche du bon caractère

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

a t a t a c

a t a t a c

a t a t a c

a t a t a c

a t a t a c

i	a	c	g	t
0	0	-	-	-
1	0	-	-	1
2	2	-	-	1
3	2	-	-	3
4	4	-	-	3
5	4	5	-	3

Une lettre du texte n'est jamais comparée plusieurs fois

Mais l'exemple est bien choisi

Parcours du texte avec la recherche du bon caractère

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

a t a t a c

a t a t a c

a t a t a c

a t a t a c

a t a t a c

i	a	c	g	t
0	0	-	-	-
1	0	-	-	1
2	2	-	-	1
3	2	-	-	3
4	4	-	-	3
5	4	5	-	3

Une lettre du texte n'est jamais comparée plusieurs fois

Mais l'exemple est bien choisi

a g a a g a g a g c

Parcours du texte avec la recherche du bon caractère

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

a t a t a c

a t a t a c

a t a t a c

a t a t a c

a t a t a c

i	a	c	g	t
0	0	-	-	-
1	0	-	-	1
2	2	-	-	1
3	2	-	-	3
4	4	-	-	3
5	4	5	-	3

Une lettre du texte n'est jamais comparée plusieurs fois

Mais l'exemple est bien choisi

a g a a g a g a g c

a g a g c

Parcours du texte avec la recherche du bon caractère

$T =$ a t a g a c a c a a t a t a c t g a c a c g a t

a t a t a c

a t a t a c

a t a t a c

a t a t a c

a t a t a c

i	a	c	g	t
0	0	-	-	-
1	0	-	-	1
2	2	-	-	1
3	2	-	-	3
4	4	-	-	3
5	4	5	-	3

Une lettre du texte n'est jamais comparée plusieurs fois

Mais l'exemple est bien choisi

a g a a g a g a g c

a g a g c

a g a g c

3- L'algorithme de Boyer Moore Horspool

3.1- Un bon décalage

Algorithme de Boyer-Moore (1977)

Algorithme de Boyer-Moore (1977)

1. Effectuer la comparaison de droite à gauche

Algorithme de Boyer-Moore (1977)

1. Effectuer la comparaison de droite à gauche

$T =$ a g a a g a g a g c

Algorithme de Boyer-Moore (1977)

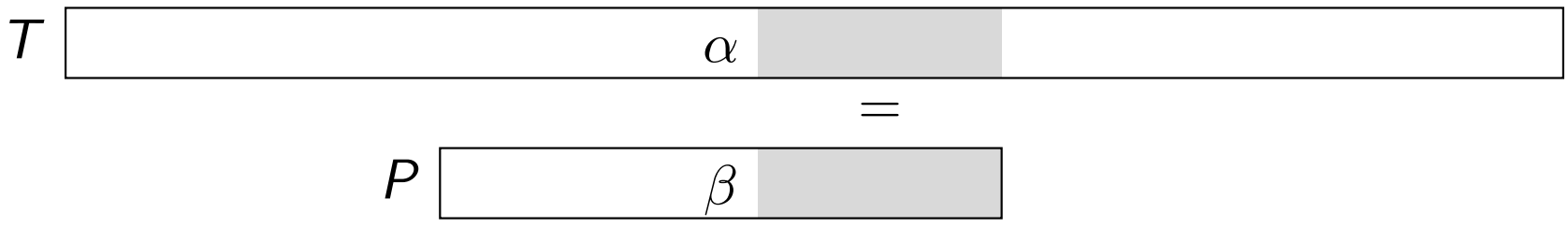
1. Effectuer la comparaison de droite à gauche

$$T = \begin{array}{cccccccccc} \mathrm{a} & \mathrm{g} & \mathrm{a} & \mathrm{a} & \mathrm{g} & \mathrm{a} & \mathrm{g} & \mathrm{a} & \mathrm{g} & \mathrm{c} \\ & & & & \mathrm{a} & \mathrm{g} & \mathrm{a} & \mathrm{g} & \overset{\times}{\mathrm{c}} & \end{array}$$

Algorithme de Boyer-Moore (1977)

1. Effectuer la comparaison de droite à gauche

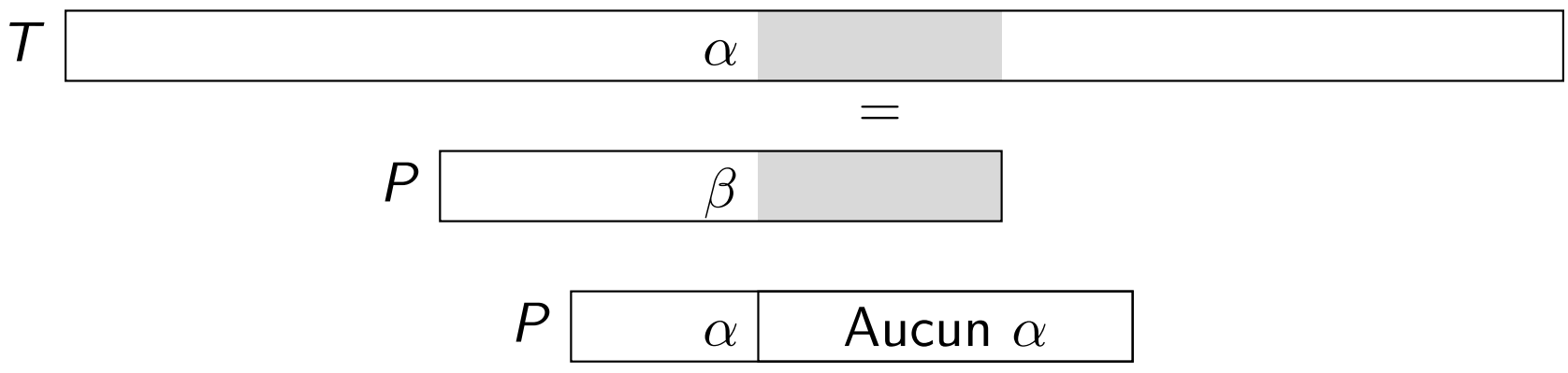
2. Utiliser la règle du bon caractère



Algorithme de Boyer-Moore (1977)

1. Effectuer la comparaison de droite à gauche

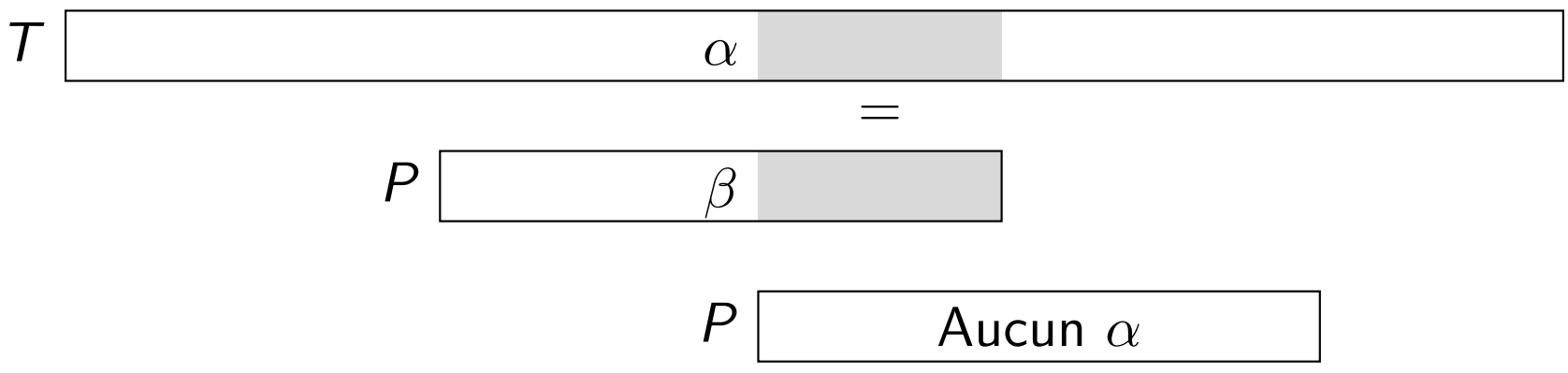
2. Utiliser la règle du bon caractère



Algorithme de Boyer-Moore (1977)

1. Effectuer la comparaison de droite à gauche

2. Utiliser la règle du bon caractère



Algorithme de Boyer-Moore (1977)

1. Effectuer la comparaison de droite à gauche

2. Utiliser la règle du bon caractère

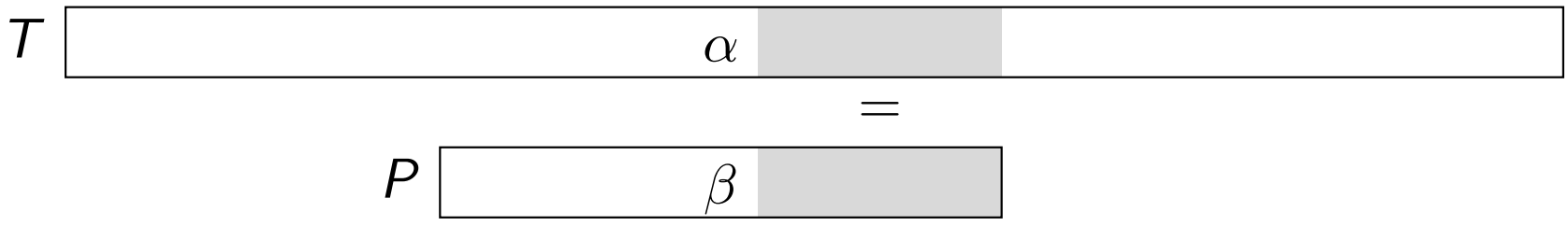
3. Utiliser la règle du bon suffixe

Algorithme de Boyer-Moore (1977)

1. Effectuer la comparaison de droite à gauche

2. Utiliser la règle du bon caractère

3. Utiliser la règle du bon suffixe

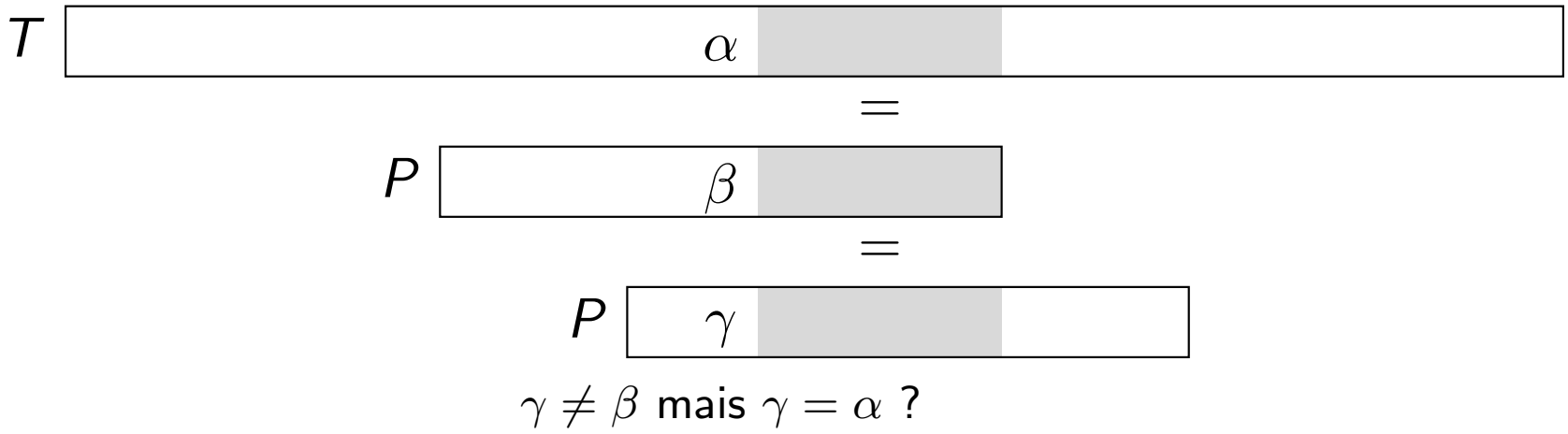


Algorithme de Boyer-Moore (1977)

1. Effectuer la comparaison de droite à gauche

2. Utiliser la règle du bon caractère

3. Utiliser la règle du bon suffixe

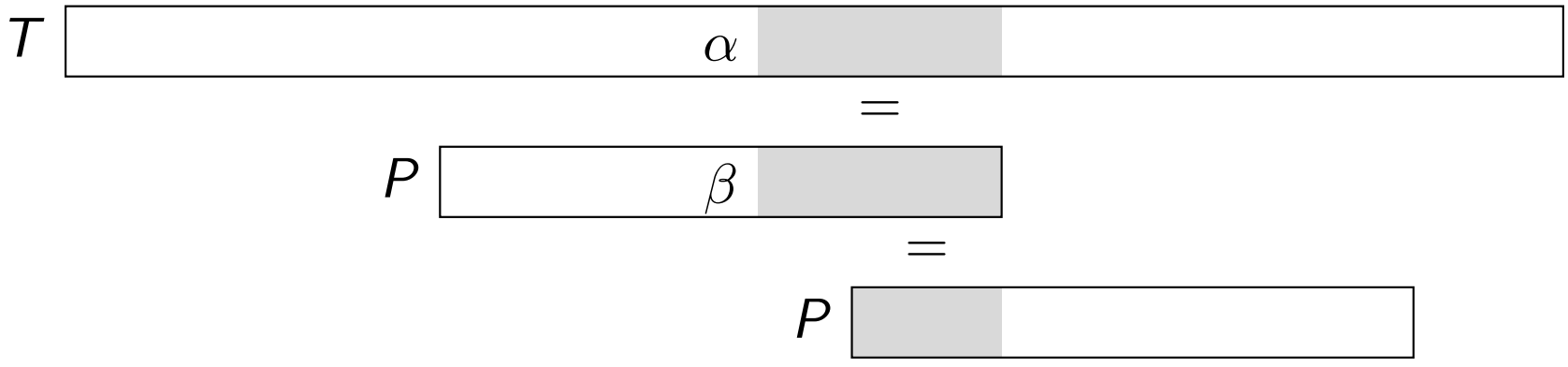


Algorithme de Boyer-Moore (1977)

1. Effectuer la comparaison de droite à gauche

2. Utiliser la règle du bon caractère

3. Utiliser la règle du bon suffixe



Synthèse sur la recherche dans un texte

Recherche dans un texte non indexé

La clé : décaler le mot recherché du plus possible

Décalage par le bon caractère, le plus long bord, le bon suffixe

Mieux vaut comparer de droite à gauche !

Quelques ressources

Explications (en anglais), exemple, code (en C) et applet Java :

- Algorithme de Knuth Morris Pratt :
  `www-igm.univ-mlv.fr/~lecroq/string/node8.html`
- Algorithme de Boyer-Moore :
  `www-igm.univ-mlv.fr/~lecroq/string/node14.html`

Chapitre 10 du livre *Éléments d'algorithmique*, avec d'autres algorithmes de recherche dans un texte en bonus :
`www-igm.univ-mlv.fr/~berstel/Elements/Elements.pdf`