

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Vers un nouvel Oracle.

Oracle à transitions gardées.

Alban MANCHERON
alban.mancheron@inria.fr

merci à Irena RUSU et Christophe MOAN

Institut National de Recherche en Informatique et en Automatique

jeudi 14 février 2008

1 Algorithmique du texte

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- 1 Algorithmique du texte
- 2 Langage reconnu par les Oracles

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- 1** Algorithmique du texte
- 2** Langage reconnu par les Oracles
- 3** Vers un nouvel Oracle

- 1 **Algorithmique du texte**
- 2 **Langage reconnu par les Oracles**
- 3 **Vers un nouvel Oracle**
- 4 **Conclusions & perspectives**

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Un mot s est une séquence finie $s = s_1 s_2 \cdots s_n$ de symboles pris dans un ensemble noté Σ .

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Un mot s est une séquence finie $s = s_1 s_2 \cdots s_n$ de symboles pris dans un ensemble noté Σ .

⇒ Lettres, Alphabet, Longueur.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Un mot s est une séquence finie $s = s_1 s_2 \cdots s_n$ de symboles pris dans un ensemble noté Σ .

⇒ Lettres, Alphabet, Longueur.

Exemple (facteurs, préfixes et suffixes)

Soit le mot CAGCTGACTAGCTACGACT.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Un mot s est une séquence finie $s = s_1 s_2 \cdots s_n$ de symboles pris dans un ensemble noté Σ .

⇒ Lettres, Alphabet, Longueur.

Exemple (facteurs, préfixes et suffixes)

Soit le mot CAGCTG**ACTAG**CTACGACT.

ACTAG est un facteur

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Un mot s est une séquence finie $s = s_1 s_2 \cdots s_n$ de symboles pris dans un ensemble noté Σ .

⇒ Lettres, Alphabet, Longueur.

Exemple (facteurs, préfixes et suffixes)

Soit le mot **CAGCT**GACTAGCTACGACT.
CAGCT est un préfixe

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Un mot s est une séquence finie $s = s_1 s_2 \cdots s_n$ de symboles pris dans un ensemble noté Σ .

⇒ Lettres, Alphabet, Longueur.

Exemple (facteurs, préfixes et suffixes)

Soit le mot CAGCTGACTAGCT**ACGACT**.

ACGACT est un suffixe

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- **Origine :**

Structure introduite en 1999 par Cyril ALLAUZEN,
Maxime CROCHEMORE et Mathieu RAFFINOT.

- **Objectif :**

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- **Origine :**

Structure introduite en 1999 par Cyril ALLAUZEN,
Maxime CROCHEMORE et Mathieu RAFFINOT.

- **Objectif :**

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- **Origine :**

Structure introduite en 1999 par Cyril ALLAUZEN,
Maxime CROCHEMORE et Mathieu RAFFINOT.

- **Objectif :**

construire un automate pour un mot s donné qui soit :

- acyclique ;

- déterministe ;

- reconnaître s en temps $O(|s|)$ et en espace $O(1)$;

- reconnaître s en temps $O(|s|)$ et en espace $O(|s|)$;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- **Origine :**

Structure introduite en 1999 par Cyril ALLAUZEN,
Maxime CROCHEMORE et Mathieu RAFFINOT.

- **Objectif :**

construire un automate pour un mot s donné qui soit :

- acyclique ;
- déterministe ;
- qui reconnaisse au moins les facteurs/suffixes de s ;
- minimal en nombre d'états ;
- linéaire en nombre de transitions.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- **Origine :**

Structure introduite en 1999 par Cyril ALLAUZEN,
Maxime CROCHEMORE et Mathieu RAFFINOT.

- **Objectif :**

construire un automate pour un mot s donné qui soit :

- acyclique ;
- déterministe ;
- qui reconnaisse au moins les facteurs/suffixes de s ;
- minimal en nombre d'états ;
- linéaire en nombre de transitions.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- **Origine :**

Structure introduite en 1999 par Cyril ALLAUZEN,
Maxime CROCHEMORE et Mathieu RAFFINOT.

- **Objectif :**

construire un automate pour un mot s donné qui soit :

- acyclique ;
- déterministe ;
- qui reconnaisse au moins les facteurs/suffixes de s ;
- minimal en nombre d'états ;
- linéaire en nombre de transitions.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- **Origine :**

Structure introduite en 1999 par Cyril ALLAUZEN,
Maxime CROCHEMORE et Mathieu RAFFINOT.

- **Objectif :**

construire un automate pour un mot s donné qui soit :

- acyclique ;
- déterministe ;
- qui reconnaisse au moins les facteurs/suffixes de s ;
- minimal en nombre d'états ;
- linéaire en nombre de transitions.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- **Origine :**

Structure introduite en 1999 par Cyril ALLAUZEN,
Maxime CROCHEMORE et Mathieu RAFFINOT.

- **Objectif :**

construire un automate pour un mot s donné qui soit :

- acyclique ;
- déterministe ;
- qui reconnaisse au moins les facteurs/suffixes de s ;
- minimal en nombre d'états ;
- linéaire en nombre de transitions.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- **Origine :**

Structure introduite en 1999 par Cyril ALLAUZEN,
Maxime CROCHEMORE et Mathieu RAFFINOT.

- **Objectif :**

construire un automate pour un mot s donné qui soit :

- acyclique ;
- déterministe ;
- qui reconnaisse au moins les facteurs/suffixes de s ;
- minimal en nombre d'états ;
- linéaire en nombre de transitions.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- **Origine :**

Structure introduite en 1999 par Cyril ALLAUZEN,
Maxime CROCHEMORE et Mathieu RAFFINOT.

- **Objectif :**

construire un automate pour un mot s donné qui soit :

- acyclique ;
- déterministe ;
- qui reconnaisse au moins les facteurs/suffixes de s ;
- minimal en nombre d'états ;
- linéaire en nombre de transitions.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- **Intérêts principaux :**

- facilité d'implémentation/d'utilisation ;
- complexités temporelle et spatiale.

- **Utilisation :**

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- **Intérêts principaux :**

- facilité d'implémentation/d'utilisation ;
- complexités temporelle et spatiale.

- **Utilisation :**

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- **Intérêts principaux :**

- facilité d'implémentation/d'utilisation ;
- complexités temporelle et spatiale.

- **Utilisation :**

- [Multi-][Turbo-]BOM (Backward Oracle Matching) :
recherche de motifs dans une ou plusieurs séquences
[ALLAUZEN & al., 2001] ;
- COMPRO : compression de textes
[LEPOTYRE & al., 2002] ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

● Intérêts principaux :

- facilité d'implémentation/d'utilisation ;
- complexités temporelle et spatiale.

● Utilisation :

- [Multi-][Turbo-]BOM (Backward Oracle Matching) : recherche de motifs dans une ou plusieurs séquences [ALLAUZEN & *al.*, 2001] ;
- COMPROR : compression de textes [LEFEBVRE & *al.*, 2002] ;
- FORREPEATS : détection de motifs répétés dans de grandes séquences [LEFEBVRE & *al.*, 2003] ;
- *OpenMusic* : improvisation musicale assistée par ordinateur [ASSAYAG & *al.*, 2004] ;
- POPITAM : identification de protéines à partir de spectrométries de masses [HERNANDEZ & *al.*, 2003].

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

● Intérêts principaux :

- facilité d'implémentation/d'utilisation ;
- complexités temporelle et spatiale.

● Utilisation :

- [Multi-][Turbo-]BOM (Backward Oracle Matching) : recherche de motifs dans une ou plusieurs séquences [ALLAUZEN & *al.*, 2001] ;
- COMPROR : compression de textes [LEFEBVRE & *al.*, 2002] ;
- FORREPEATS : détection de motifs répétés dans de grandes séquences [LEFEBVRE & *al.*, 2003] ;
- *OpenMusic* : improvisation musicale assistée par ordinateur [ASSAYAG & *al.*, 2004] ;
- POPITAM : identification de protéines à partir de spectrométries de masses [HERNANDEZ & *al.*, 2003].

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

● Intérêts principaux :

- facilité d'implémentation/d'utilisation ;
- complexités temporelle et spatiale.

● Utilisation :

- [Multi-][Turbo-]BOM (Backward Oracle Matching) : recherche de motifs dans une ou plusieurs séquences [ALLAUZEN & *al.*, 2001] ;
- COMPROR : compression de textes [LEFEBVRE & *al.*, 2002] ;
- FORREPEATS : détection de motifs répétés dans de grandes séquences [LEFEBVRE & *al.*, 2003] ;
- *OpenMusic* : improvisation musicale assistée par ordinateur [ASSAYAG & *al.*, 2004] ;
- POPITAM : identification de protéines à partir de spectrométries de masses [HERNANDEZ & *al.*, 2003].

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

● Intérêts principaux :

- facilité d'implémentation/d'utilisation ;
- complexités temporelle et spatiale.

● Utilisation :

- [Multi-][Turbo-]BOM (Backward Oracle Matching) : recherche de motifs dans une ou plusieurs séquences [ALLAUZEN & *al.*, 2001] ;
- COMPROR : compression de textes [LEFEBVRE & *al.*, 2002] ;
- FORREPEATS : détection de motifs répétés dans de grandes séquences [LEFEBVRE & *al.*, 2003] ;
- *OpenMusic* : improvisation musicale assistée par ordinateur [ASSAYAG & *al.*, 2004] ;
- POPITAM : identification de protéines à partir de spectrométries de masses [HERNANDEZ & *al.*, 2003].

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

● Intérêts principaux :

- facilité d'implémentation/d'utilisation ;
- complexités temporelle et spatiale.

● Utilisation :

- [Multi-][Turbo-]BOM (Backward Oracle Matching) : recherche de motifs dans une ou plusieurs séquences [ALLAUZEN & *al.*, 2001] ;
- COMPROR : compression de textes [LEFEBVRE & *al.*, 2002] ;
- FORREPEATS : détection de motifs répétés dans de grandes séquences [LEFEBVRE & *al.*, 2003] ;
- *OpenMusic* : improvisation musicale assistée par ordinateur [ASSAYAG & *al.*, 2004] ;
- POPITAM : identification de protéines à partir de spectrométries de masses [HERNANDEZ & *al.*, 2003].

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

● Intérêts principaux :

- facilité d'implémentation/d'utilisation ;
- complexités temporelle et spatiale.

● Utilisation :

- [Multi-][Turbo-]BOM (Backward Oracle Matching) : recherche de motifs dans une ou plusieurs séquences [ALLAUZEN & *al.*, 2001] ;
- COMPROR : compression de textes [LEFEBVRE & *al.*, 2002] ;
- FORREPEATS : détection de motifs répétés dans de grandes séquences [LEFEBVRE & *al.*, 2003] ;
- *OpenMusic* : improvisation musicale assistée par ordinateur [ASSAYAG & *al.*, 2004] ;
- POPITAM : identification de protéines à partir de spectrométries de masses [HERNANDEZ & *al.*, 2003].

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

```
1 Entrées :  $\Sigma$  % Alphabet (supposé minimal) %
2            $s \in \Sigma^*$  % Le mot à traiter %
3 Sortie : Oracle % L'Oracle des Facteurs de  $s$  %
4
5 Début
6   Créer l'état initial étiqueté par  $e_0$ 
7
8   Pour  $i$  de 1 à  $|s|$  Faire
9     Créer un état étiqueté par  $e_i$ 
10    Ajouter une transition de l'état  $e_{i-1}$  vers l'état  $e_i$  étiquetée par  $s[i]$ 
11  Fin Pour
12
13  Pour  $i$  de 0 à  $|s| - 1$  Faire
14    Soit  $u$  un mot de longueur minimale reconnu à l'état  $e_i$ 
15    Pour Tout  $\alpha \in \Sigma \setminus \{s[i+1]\}$  Faire
16      Si  $u\alpha \in \text{Fact}(s[i - |u| + 1..|s|])$  Alors
17         $j \leftarrow \text{poccur}_{s[i - |u| + 1..|s|]}(u\alpha) - |u|$ 
18        Ajouter une transition de l'état  $e_i$  vers l'état  $e_{i+j}$  étiquetée par  $\alpha$ 
19      Fin Si
20    Fin Pour Tout
21  Fin Pour
22 Fin
```

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

```
1 Entrées :  $\Sigma$  % Alphabet (supposé minimal) %  
2            $s \in \Sigma^*$  % Le mot à traiter %  
3 Sortie : Oracle % L'Oracle de  $s$  %  
4  
5 Début  
6   Créer l'état initial étiqueté par  $e_0$   
7    $\mathcal{S}_s(e_0) \leftarrow e_{-1}$   
8  
9   Pour  $i$  de 1 à  $|s|$  Faire  
10  Créer un état étiqueté par  $e_i$   
11  Ajouter une transition de l'état  $e_{i-1}$  vers l'état  $e_i$  étiquetée par  $s[i]$   
12   $e_k \leftarrow \mathcal{S}_s(e_{i-1})$   
13  Tant Que  $k > -1$  et  
14  qu'il n'existe pas de transition issue de  $e_k$  étiquetée par  $s[i]$  Faire  
15  Ajouter une transition de l'état  $e_k$  vers l'état  $e_i$  étiquetée par  $s[i]$   
16   $e_k \leftarrow \mathcal{S}_s(e_k)$   
17  Fin Tant Que  
18  Si  $(k = -1)$  Alors  
19   $\mathcal{S}_s(e_i) \leftarrow e_0$   
20  Sinon  
21  Soit  $e_x$  l'état d'arrivée de la transition issue de  $e_k$  étiquetée par  $s[i]$   
22   $\mathcal{S}_s(e_i) \leftarrow e_x$   
23  Fin Si  
24  Fin Pour  
25  Fin
```

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Oracle des Suffixes du mot *gaccattctc*



*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 1 \quad s[i + 1] = g$$

$$\mathcal{S}_\varepsilon(0) = -1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

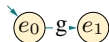
Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 1 \quad s[i + 1] = g$$

$$\mathcal{S}_\varepsilon(0) = -1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

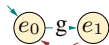
Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 2 \quad s[i + 1] = a$$

$$\mathcal{S}_g(1) = 0$$

$$\mathcal{S}_g(0) = -1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

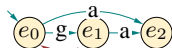
Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 2 \quad s[i + 1] = a$$

$$\mathcal{S}_g(1) = 0$$

$$\mathcal{S}_g(0) = -1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

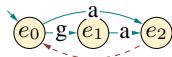
Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 3 \quad s[i + 1] = c$$

$$\mathcal{S}_{ga}(2) = 0$$

$$\mathcal{S}_{ga}(0) = -1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

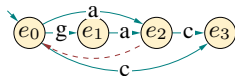
Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 3 \quad s[i + 1] = c$$

$$\mathcal{S}_{ga}(2) = 0$$

$$\mathcal{S}_{ga}(0) = -1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

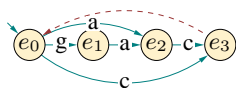
Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 4 \quad s[i + 1] = c$$

$$\mathcal{S}_{gac}(3) = 0$$

$$\mathcal{S}_{gac}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

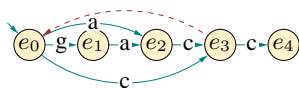
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 4 \quad s[i + 1] = c$$

$$\mathcal{S}_{gac}(3) = 0$$

$$\mathcal{S}_{gac}(0) = -1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

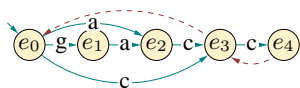
Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 5 \quad s[i + 1] = a$$

$$\mathcal{S}_{gacc}(4) = 3$$

$$\mathcal{S}_{gacc}(3) = 0$$

$$\mathcal{S}_{gacc}(0) = -1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

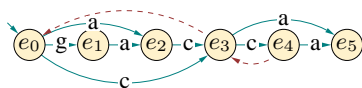
Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 5 \quad s[i + 1] = a$$

$$\mathcal{S}_{gacc}(4) = 3$$

$$\mathcal{S}_{gacc}(3) = 0$$

$$\mathcal{S}_{gacc}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

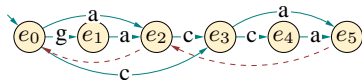
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 6 \quad s[i + 1] = t$$

$$\mathcal{S}_{gacca}(5) = 2$$

$$\mathcal{S}_{gacca}(2) = 0$$

$$\mathcal{S}_{gacca}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

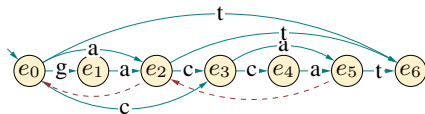
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 6 \quad s[i + 1] = t$$

$$\mathcal{S}_{gacca}(5) = 2$$

$$\mathcal{S}_{gacca}(2) = 0$$

$$\mathcal{S}_{gacca}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

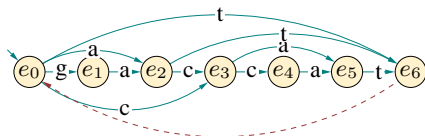
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 7 \quad s[i + 1] = t$$

$$\mathcal{S}_{gaccat}(6) = 0$$

$$\mathcal{S}_{gaccat}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

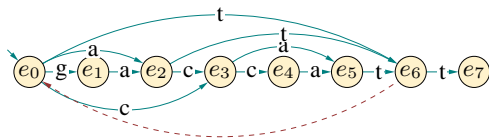
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 7 \quad s[i + 1] = t$$

$$\mathcal{S}_{gaccat}(6) = 0$$

$$\mathcal{S}_{gaccat}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

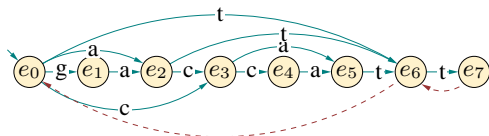
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 8 \quad s[i + 1] = c$$

$$\mathcal{S}_{gaccatt}(7) = 6$$

$$\mathcal{S}_{gaccatt}(6) = 0$$

$$\mathcal{S}_{gaccatt}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

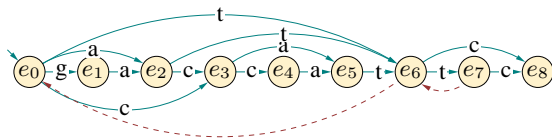
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 8 \quad s[i + 1] = c$$

$$\mathcal{S}_{gaccatt}(7) = 6$$

$$\mathcal{S}_{gaccatt}(6) = 0$$

$$\mathcal{S}_{gaccatt}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

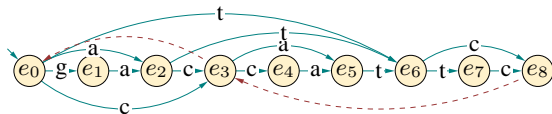
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 9 \quad s[i + 1] = t$$

$$\mathcal{S}_{gaccattc}(8) = 3$$

$$\mathcal{S}_{gaccattc}(3) = 0$$

$$\mathcal{S}_{gaccattc}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

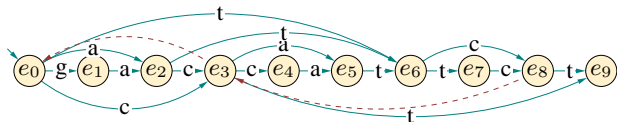
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 9 \quad s[i + 1] = t$$

$$\mathcal{S}_{gaccattc}(8) = 3$$

$$\mathcal{S}_{gaccattc}(3) = 0$$

$$\mathcal{S}_{gaccattc}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

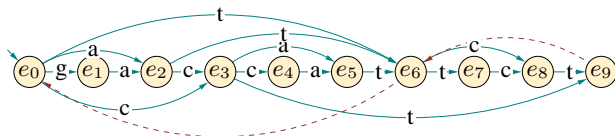
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 10 \quad s[i + 1] = c$$

$$\mathcal{S}_{gaccattctc}(9) = 6$$

$$\mathcal{S}_{gaccattctc}(6) = 0$$

$$\mathcal{S}_{gaccattctc}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

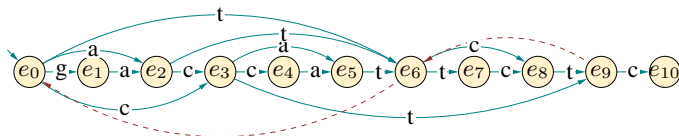
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Oracle des Suffixes du mot *gaccattctc*



$$i + 1 = 10 \quad s[i + 1] = c$$

$$\mathcal{S}_{gaccattctc}(9) = 6$$

$$\mathcal{S}_{gaccattctc}(6) = 0$$

$$\mathcal{S}_{gaccattctc}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

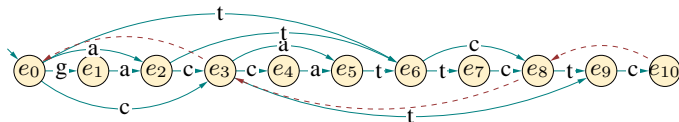
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Oracle des Suffixes du mot *gaccattctc*



$$\mathcal{S}_{gaccattctc}(10) = 8$$

$$\mathcal{S}_{gaccattctc}(8) = 3$$

$$\mathcal{S}_{gaccattctc}(3) = 0$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

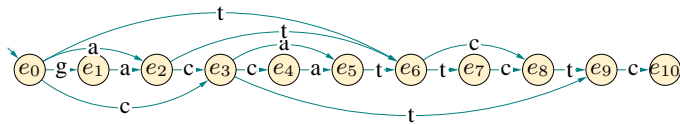
Vers un nouvel Oracle

*Conclusions &
perspectives*

Oracles

Facteurs

Suffixes



*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

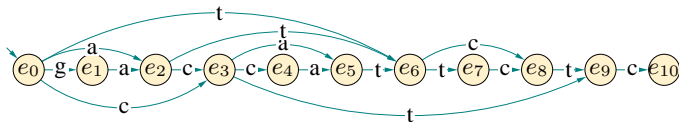
*Conclusions &
perspectives*

Oracles

Facteurs

Suffixes

Tous les états sont
finaux



*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

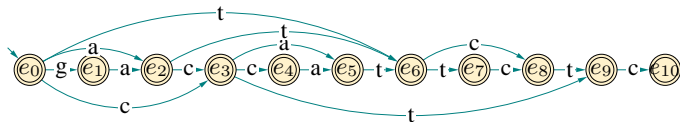
*Conclusions &
perspectives*

Oracles

Facteurs

Suffixes

Tous les états sont
finaux



*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

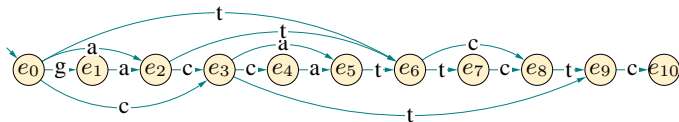
Oracles

Facteurs

Suffixes

Tous les états sont
finaux

Seuls les états e_i tels
qu'il existe un chemin
étiquetant un suffixe de
 s sont terminaux



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

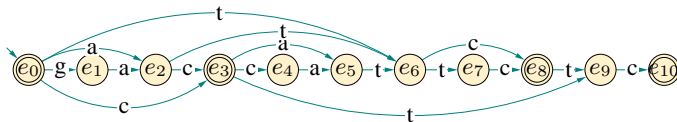
Oracles

Facteurs

Suffixes

Tous les états sont
finaux

Seuls les états e_i tels
qu'il existe un chemin
étiquetant un suffixe de
 s sont terminaux



*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

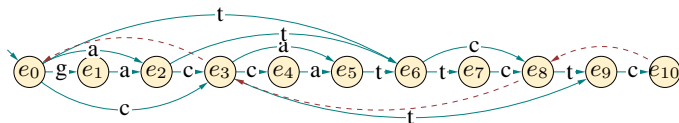
Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Oracle des Suffixes du mot *gaccattctc*



$$\mathcal{S}_{gaccattctc}(10) = 8$$

$$\mathcal{S}_{gaccattctc}(8) = 3$$

$$\mathcal{S}_{gaccattctc}(3) = 0$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

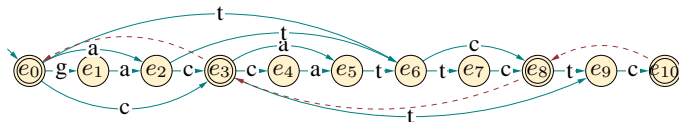
Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Oracle des Suffixes du mot *gaccattctc*



*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

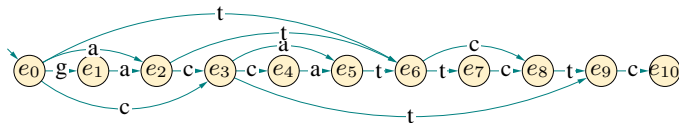
Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Oracle des Facteurs du mot *gaccattctc*



*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

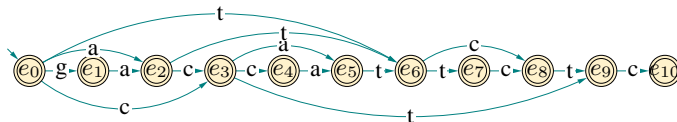
Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Oracle des Facteurs du mot *gaccattctc*



*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Propriété

Étant donné un mot $s \in \Sigma^+$ de longueur n , le nombre d'état de l'Oracle de s est exactement $n + 1$ et le nombre de transitions est compris entre n et $2n - 1$.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Propriété

Étant donné un mot $s \in \Sigma^+$ de longueur n , le nombre d'état de l'Oracle de s est exactement $n + 1$ et le nombre de transitions est compris entre n et $2n - 1$.

- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Propriété

L'Oracle est un automate homogène – i.e., toutes les transitions arrivant dans un même état sont étiquetées par le même symbole.

- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Propriété

L'Oracle est un automate homogène – i.e., toutes les transitions arrivant dans un même état sont étiquetées par le même symbole.

- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Définition

Étant donné un mot $s \in \Sigma^+$ et un facteur x de s , la fonction \underline{Pos} est définie comme étant la position de la première occurrence de x dans s , *i.e.*, s peut être écrit $s = u x v$ ($u, v \in \Sigma^*$, $x \notin Fact(s[1..|u x| - 1])$) et alors $Pos_s(x) = |u| + 1$. La fonction \underline{poccur} est alors définie telle que $poccur_s(x) = |u x| = Pos_s(x) + |x| - 1$.

- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Définition

Étant donné un mot $s \in \Sigma^+$ et un facteur x de s , la fonction \underline{Pos} est définie comme étant la position de la première occurrence de x dans s , *i.e.*, s peut être écrit $s = u x v$ ($u, v \in \Sigma^*$, $x \notin \text{Fact}(s[1..|u x| - 1])$) et alors $\text{Pos}_s(x) = |u| + 1$. La fonction \underline{poccur} est alors définie telle que $\text{poccur}_s(x) = |u x| = \text{Pos}_s(x) + |x| - 1$.

- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $\text{Pos}(u, s)$ est la position du début de la première occurrence de u dans s ;
- $\text{poccur}(u, s)$ est la position de la fin de la première occurrence de u dans s ($= \text{Pos}(u, s) + |u| - 1$) ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Propriété

Étant donné un mot $s \in \Sigma^+$, l'Oracle des Facteurs de s reconnaît au moins tous les facteurs de s et l'Oracle des Suffixes de s reconnaît au moins tous les suffixes de s .

- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $Pos(u, s)$ est la position du début de la première occurrence de u dans s ;
- $poocur(u, s)$ est la position de la fin de la première occurrence de u dans s ($= Pos(u, s) + |u| - 1$) ;

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Propriété

Étant donné un mot $s \in \Sigma^+$, l'Oracle des Facteurs de s reconnaît au moins tous les facteurs de s et l'Oracle des Suffixes de s reconnaît au moins tous les suffixes de s .

- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $Pos(u, s)$ est la position du début de la première occurrence de u dans s ;
- $poocur(u, s)$ est la position de la fin de la première occurrence de u dans s ($= Pos(u, s) + |u| - 1$) ;
- l'Oracle des Facteurs reconnaît au moins $Fact(s)$;
- l'Oracle des Suffixes reconnaît au moins $Suff(s)$;

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Lemme

Étant donné un mot $s \in \Sigma^*$ et son Oracle, un mot unique de longueur minimum est accepté à chaque état e_i ($0 \leq i \leq |s|$) de l'Oracle de s . Il est noté $\min(e_i)$. De plus, pour tout entier i ($0 \leq i \leq |s|$), $\min(e_i) \in \text{Fact}(s)$ et $i = \text{poccur}_s(\min(e_i))$.

- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $\text{Pos}(u, s)$ est la position du début de la première occurrence de u dans s ;
- $\text{poccur}(u, s)$ est la position de la fin de la première occurrence de u dans s ($= \text{Pos}(u, s) + |u| - 1$) ;
- l'Oracle des Facteurs reconnaît au moins $\text{Fact}(s)$;
- l'Oracle des Suffixes reconnaît au moins $\text{Suff}(s)$;

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Lemme

Étant donné un mot $s \in \Sigma^*$ et son Oracle, un mot unique de longueur minimum est accepté à chaque état e_i ($0 \leq i \leq |s|$) de l'Oracle de s . Il est noté $\min(e_i)$. De plus, pour tout entier i ($0 \leq i \leq |s|$), $\min(e_i) \in \text{Fact}(s)$ et $i = \text{poccur}_s(\min(e_i))$.

- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $\text{Pos}(u, s)$ est la position du début de la première occurrence de u dans s ;
- $\text{poccur}(u, s)$ est la position de la fin de la première occurrence de u dans s ($= \text{Pos}(u, s) + |u| - 1$) ;
- l'Oracle des Facteurs reconnaît au moins $\text{Fact}(s)$;
- l'Oracle des Suffixes reconnaît au moins $\text{Suff}(s)$;
- il n'existe qu'un seul mot de taille minimale reconnu à chaque état : $\min(e_i) \in \text{Fact}(s)$.
 $\text{poccur}_s(\min(e_i)) = i$;

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Lemme

Étant donné un mot $s \in \Sigma^*$ et un entier i ($0 \leq i \leq |s|$), le mot $\text{min}(e_i)$ est suffixe de tout mot reconnu à l'état e_i de l'Oracle de s .

- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $\text{Pos}(u, s)$ est la position du début de la première occurrence de u dans s ;
- $\text{poccur}(u, s)$ est la position de la fin de la première occurrence de u dans s ($= \text{Pos}(u, s) + |u| - 1$) ;
- l'Oracle des Facteurs reconnaît au moins $\text{Fact}(s)$;
- l'Oracle des Suffixes reconnaît au moins $\text{Suff}(s)$;
- il n'existe qu'un seul mot de taille minimale reconnu à chaque état : $\text{min}(e_i) \in \text{Fact}(s)$.
 $\text{poccur}_s(\text{min}(e_i)) = i$;

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Lemme

Étant donné un mot $s \in \Sigma^*$ et un entier i ($0 \leq i \leq |s|$), le mot $\text{min}(e_i)$ est suffixe de tout mot reconnu à l'état e_i de l'Oracle de s .

- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $\text{Pos}(u, s)$ est la position du début de la première occurrence de u dans s ;
- $\text{poccur}(u, s)$ est la position de la fin de la première occurrence de u dans s ($= \text{Pos}(u, s) + |u| - 1$) ;
- l'Oracle des Facteurs reconnaît au moins $\text{Fact}(s)$;
- l'Oracle des Suffixes reconnaît au moins $\text{Suff}(s)$;
- il n'existe qu'un seul mot de taille minimale reconnu à chaque état : $\text{min}(e_i) \in \text{Fact}(s)$.
 $\text{poccur}_s(\text{min}(e_i)) = i$;
- $\text{min}(e_i)$ est suffixe de tous les mots reconnus à l'état e_i ;

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Définition

Étant donné un mot $s \in \Sigma^*$ et un mot w reconnu par l'Oracle de s à l'état e_i ($0 \leq i \leq |s|$), la fonction Etat est définie comme étant l'état de reconnaissance du mot w par l'Oracle de s ; i.e., $Etat(w) = e_i$.

- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $Pos(u, s)$ est la position du début de la première occurrence de u dans s ;
- $poccur(u, s)$ est la position de la fin de la première occurrence de u dans s ($= Pos(u, s) + |u| - 1$) ;
- l'Oracle des Facteurs reconnaît au moins $Fact(s)$;
- l'Oracle des Suffixes reconnaît au moins $Suff(s)$;
- il n'existe qu'un seul mot de taille minimale reconnu à chaque état : $min(e_i) \in Fact(s)$.
 $poccur_s(min(e_i)) = i$;
- $min(e_i)$ est suffixe de tous les mots reconnus à l'état e_i ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Définition

Étant donné un mot $s \in \Sigma^*$ et un mot w reconnu par l'Oracle de s à l'état e_i ($0 \leq i \leq |s|$), la fonction Etat est définie comme étant l'état de reconnaissance du mot w par l'Oracle de s ; i.e., $Etat(w) = e_i$.

- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $Pos(u, s)$ est la position du début de la première occurrence de u dans s ;
- $poccur(u, s)$ est la position de la fin de la première occurrence de u dans s ($= Pos(u, s) + |u| - 1$) ;
- l'Oracle des Facteurs reconnaît au moins $Fact(s)$;
- l'Oracle des Suffixes reconnaît au moins $Suff(s)$;
- il n'existe qu'un seul mot de taille minimale reconnu à chaque état : $min(e_i) \in Fact(s)$.
 $poccur_s(min(e_i)) = i$;
- $min(e_i)$ est suffixe de tous les mots reconnus à l'état e_i ;
- un mot w reconnu par l'Oracle de s l'est à l'état $e_i = Etat(w)$;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Notation

Étant donné un automate et e_i un état de cet automate, le degré entrant de e_i (i.e., le nombre de transitions arrivant à l'état e_i) est noté $\#_{in}(e_i)$ et le degré sortant de e_i (i.e., le nombre de transitions issues de l'état e_i) est noté $\#_{out}(e_i)$.

- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $Pos(u, s)$ est la position du début de la première occurrence de u dans s ;
- $poccur(u, s)$ est la position de la fin de la première occurrence de u dans s ($= Pos(u, s) + |u| - 1$) ;
- l'Oracle des Facteurs reconnaît au moins $Fact(s)$;
- l'Oracle des Suffixes reconnaît au moins $Suff(s)$;
- il n'existe qu'un seul mot de taille minimale reconnu à chaque état : $min(e_i) \in Fact(s)$.
 $poccur_s(min(e_i)) = i$;
- $min(e_i)$ est suffixe de tous les mots reconnus à l'état e_i ;
- un mot w reconnu par l'Oracle de s l'est à l'état $e_i = Etat(w)$;

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

Notation

Étant donné un automate et e_i un état de cet automate, le degré entrant de e_i (i.e., le nombre de transitions arrivant à l'état e_i) est noté $\#_{in}(e_i)$ et le degré sortant de e_i (i.e., le nombre de transitions issues de l'état e_i) est noté $\#_{out}(e_i)$.

- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $Pos(u, s)$ est la position du début de la première occurrence de u dans s ;
- $poccur(u, s)$ est la position de la fin de la première occurrence de u dans s ($= Pos(u, s) + |u| - 1$) ;
- l'Oracle des Facteurs reconnaît au moins $Fact(s)$;
- l'Oracle des Suffixes reconnaît au moins $Suff(s)$;
- il n'existe qu'un seul mot de taille minimale reconnu à chaque état : $min(e_i) \in Fact(s)$.
 $poccur_s(min(e_i)) = i$;
- $min(e_i)$ est suffixe de tous les mots reconnus à l'état e_i ;
- un mot w reconnu par l'Oracle de s l'est à l'état $e_i = Etat(w)$;
- $\#_{in}(e_i)$ et $\#_{out}(e_i)$ désignent respectivement le nombre de transitions entrantes et sortantes de l'état e_i .

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

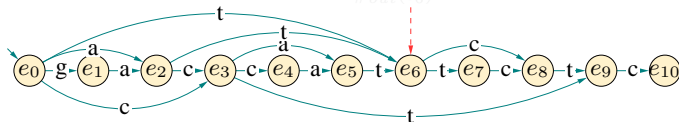
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

$Etat(cat) = e_6$
 $min(e_6) = t$
 $\#_{out}(e_6) = 2$



- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $Pos(u, s)$ est la position du début de la première occurrence de u dans s ;
- $poccur(u, s)$ est la position de la fin de la première occurrence de u dans s ($= Pos(u, s) + |u| - 1$) ;
- l'Oracle des Facteurs reconnaît au moins $Fact(s)$;
- l'Oracle des Suffixes reconnaît au moins $Suff(s)$;
- il n'existe qu'un seul mot de taille minimale reconnu à chaque état : $min(e_i) \in Fact(s)$.
 $poccur_s(min(e_i)) = i$;
- $min(e_i)$ est suffixe de tous les mots reconnus à l'état e_i ;
- un mot w reconnu par l'Oracle de s l'est à l'état $e_i = Etat(w)$;
- $\#_{in}(e_i)$ et $\#_{out}(e_i)$ désignent respectivement le nombre de transitions entrantes et sortantes de l'état e_i .

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

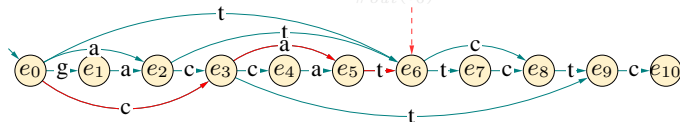
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

$Etat(cat) = e_6$
 $min(e_6) = t$
 $\#_{out}(e_6) = 2$



- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $Pos(u, s)$ est la position du début de la première occurrence de u dans s ;
- $poccur(u, s)$ est la position de la fin de la première occurrence de u dans s ($= Pos(u, s) + |u| - 1$) ;
- l'Oracle des Facteurs reconnaît au moins $Fact(s)$;
- l'Oracle des Suffixes reconnaît au moins $Suff(s)$;
- il n'existe qu'un seul mot de taille minimale reconnu à chaque état : $min(e_i) \in Fact(s)$.
 $poccur_s(min(e_i)) = i$;
- $min(e_i)$ est suffixe de tous les mots reconnus à l'état e_i ;
- un mot w reconnu par l'Oracle de s l'est à l'état $e_i = Etat(w)$;
- $\#_{in}(e_i)$ et $\#_{out}(e_i)$ désignent respectivement le nombre de transitions entrantes et sortantes de l'état e_i .

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

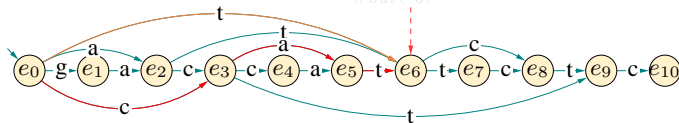
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

$Etat(cat) = e_6$
 $min(e_6) = t$
 $\#_{out}(e_6) = 2$



- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $Pos(u, s)$ est la position du début de la première occurrence de u dans s ;
- $poccur(u, s)$ est la position de la fin de la première occurrence de u dans s ($= Pos(u, s) + |u| - 1$) ;
- l'Oracle des Facteurs reconnaît au moins $Fact(s)$;
- l'Oracle des Suffixes reconnaît au moins $Suff(s)$;
- il n'existe qu'un seul mot de taille minimale reconnu à chaque état : $min(e_i) \in Fact(s)$.
 $poccur_s(min(e_i)) = i$;
- $min(e_i)$ est suffixe de tous les mots reconnus à l'état e_i ;
- un mot w reconnu par l'Oracle de s l'est à l'état $e_i = Etat(w)$;
- $\#_{in}(e_i)$ et $\#_{out}(e_i)$ désignent respectivement le nombre de transitions entrantes et sortantes de l'état e_i .

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

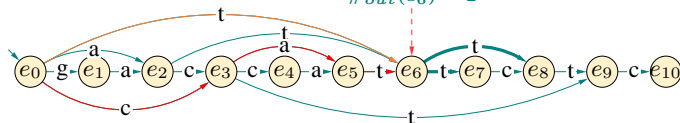
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

$Etat(cat) = e_6$
 $min(e_6) = t$
 $\#out(e_6) = 2$



- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $Pos(u, s)$ est la position du début de la première occurrence de u dans s ;
- $poccur(u, s)$ est la position de la fin de la première occurrence de u dans s ($= Pos(u, s) + |u| - 1$) ;
- l'Oracle des Facteurs reconnaît au moins $Fact(s)$;
- l'Oracle des Suffixes reconnaît au moins $Suff(s)$;
- il n'existe qu'un seul mot de taille minimale reconnu à chaque état : $min(e_i) \in Fact(s)$.
 $poccur_s(min(e_i)) = i$;
- $min(e_i)$ est suffixe de tous les mots reconnus à l'état e_i ;
- un mot w reconnu par l'Oracle de s l'est à l'état $e_i = Etat(w)$;
- $\#in(e_i)$ et $\#out(e_i)$ désignent respectivement le nombre de transitions entrantes et sortantes de l'état e_i .

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

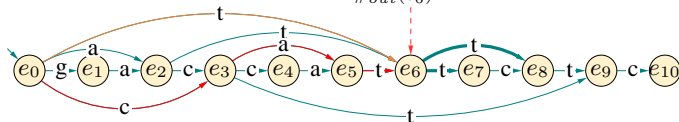
Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

$$\begin{aligned} \text{Etat}(\text{cat}) &= e_6 \\ \min(e_6) &= t \\ \#_{\text{out}}(e_6) &= 2 \end{aligned}$$



- l'Oracle compte exactement $n + 1$ états ;
- l'Oracle compte entre n et $2n - 1$ arcs ;
- l'Oracle est homogène ;
- $Pos(u, s)$ est la position du début de la première occurrence de u dans s ;
- $poccur(u, s)$ est la position de la fin de la première occurrence de u dans s ($= Pos(u, s) + |u| - 1$) ;
- l'Oracle des Facteurs reconnaît au moins $Fact(s)$;
- l'Oracle des Suffixes reconnaît au moins $Suff(s)$;
- il n'existe qu'un seul mot de taille minimale reconnu à chaque état : $\min(e_i) \in Fact(s)$.
 $poccur_s(\min(e_i)) = i$;
- $\min(e_i)$ est suffixe de tous les mots reconnus à l'état e_i ;
- un mot w reconnu par l'Oracle de s l'est à l'état $e_i = \text{Etat}(w)$;
- $\#_{in}(e_i)$ et $\#_{out}(e_i)$ désignent respectivement le nombre de transitions entrantes et sortantes de l'état e_i .

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

- L'Oracle des Facteurs/Suffixes de s peut reconnaître des mots qui ne sont pas facteurs/suffixes de s .

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

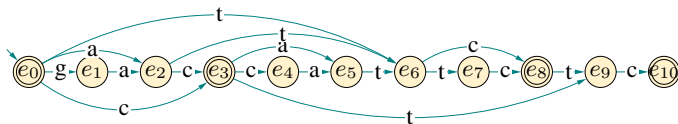
*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- L'Oracle des Facteurs/Suffixes de s peut reconnaître des mots qui ne sont pas facteurs/suffixes de s .

$atc \notin \text{Fact}(gaccattctc)$
 $\notin \text{Suff}(gaccattctc)$



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

Difficulté d'utilisation

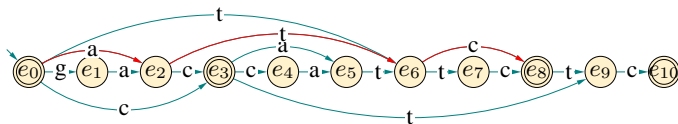
Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

- L'Oracle des Facteurs/Suffixes de s peut reconnaître des mots qui ne sont pas facteurs/suffixes de s .

$atc \notin \text{Fact}(gaccattctc)$
 $\notin \text{Suff}(gaccattctc)$



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

Oracle des Facteurs vs. Oracle des
Suffixes

Principales propriétés

Difficulté d'utilisation

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Conclusions &
perspectives

- L'Oracle des Facteurs/Suffixes de s peut reconnaître des mots qui ne sont pas facteurs/suffixes de s .
- L'Oracle des Facteurs/Suffixes n'est pas minimal en nombre de mots reconnus dans l'ensemble des automates homogènes à $n + 1$ états reconnaissant au moins les facteurs/suffixes de s .

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

- L'Oracle des Facteurs/Suffixes de s peut reconnaître des mots qui ne sont pas facteurs/suffixes de s .
 - L'Oracle des Facteurs/Suffixes n'est pas minimal en nombre de mots reconnus dans l'ensemble des automates homogènes à $n + 1$ états reconnaissant au moins les facteurs/suffixes de s .
- ⇒ Besoin de caractériser le langage reconnu par les Oracles !

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Rappels

Introduction aux Oracles

Algorithme « off-line »

Algorithme « on-line »

Un premier exemple

*Oracle des Facteurs vs. Oracle des
Suffixes*

Principales propriétés

Difficulté d'utilisation

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Définition (Facteurs Remarquables)

Étant donné un mot $s \in \Sigma^*$ et son Oracle, l'ensemble des facteurs remarquables de s , noté \mathcal{F}_s , est défini par :

$$\mathcal{F}_s = \{\min(e_i) \mid 1 \leq i < |s| \wedge (\#_{out}(e_i) > 1 \vee \min(e_i) \in Suff(s))\},$$

où $\#_{out}(e_i)$ désigne le degré sortant de l'état e_i .

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

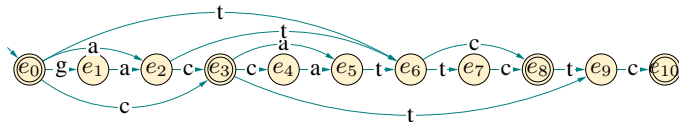
*Conclusions &
perspectives*

Définition (Facteurs Remarquables)

Étant donné un mot $s \in \Sigma^*$ et son Oracle, l'ensemble des facteurs remarquables de s , noté \mathcal{F}_s , est défini par :

$$\mathcal{F}_s = \{ \min(e_i) \mid 1 \leq i < |s| \wedge (\#_{out}(e_i) > 1 \vee \min(e_i) \in Suff(s)) \},$$

où $\#_{out}(e_i)$ désigne le degré sortant de l'état e_i .



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

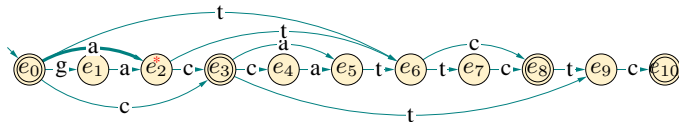
Conclusions &
perspectives

Définition (Facteurs Remarquables)

Étant donné un mot $s \in \Sigma^*$ et son Oracle, l'ensemble des facteurs remarquables de s , noté \mathcal{F}_s , est défini par :

$$\mathcal{F}_s = \{ \min(e_i) \mid 1 \leq i < |s| \wedge (\#_{out}(e_i) > 1 \vee \min(e_i) \in Suff(s)) \},$$

où $\#_{out}(e_i)$ désigne le degré sortant de l'état e_i .



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

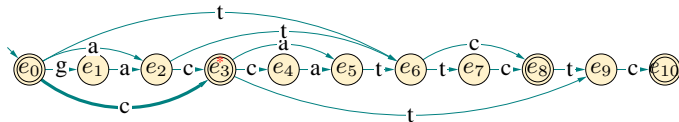
Conclusions &
perspectives

Définition (Facteurs Remarquables)

Étant donné un mot $s \in \Sigma^*$ et son Oracle, l'ensemble des facteurs remarquables de s , noté \mathcal{F}_s , est défini par :

$$\mathcal{F}_s = \{ \min(e_i) \mid 1 \leq i < |s| \wedge (\#_{out}(e_i) > 1 \vee \min(e_i) \in Suff(s)) \},$$

où $\#_{out}(e_i)$ désigne le degré sortant de l'état e_i .



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

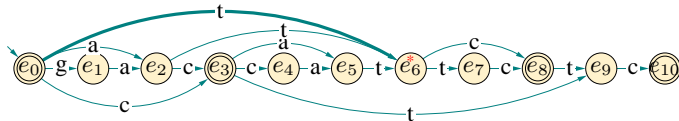
Conclusions &
perspectives

Définition (Facteurs Remarquables)

Étant donné un mot $s \in \Sigma^*$ et son Oracle, l'ensemble des facteurs remarquables de s , noté \mathcal{F}_s , est défini par :

$$\mathcal{F}_s = \{ \min(e_i) \mid 1 \leq i < |s| \wedge (\#_{out}(e_i) > 1 \vee \min(e_i) \in Suff(s)) \},$$

où $\#_{out}(e_i)$ désigne le degré sortant de l'état e_i .



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

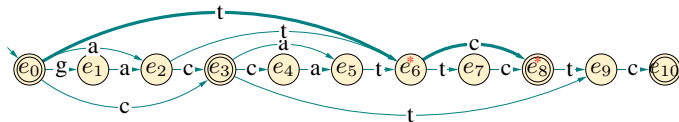
Conclusions &
perspectives

Définition (Facteurs Remarquables)

Étant donné un mot $s \in \Sigma^*$ et son Oracle, l'ensemble des facteurs remarquables de s , noté \mathcal{F}_s , est défini par :

$$\mathcal{F}_s = \{ \min(e_i) \mid 1 \leq i < |s| \wedge (\#_{out}(e_i) > 1 \vee \min(e_i) \in Suff(s)) \},$$

où $\#_{out}(e_i)$ désigne le degré sortant de l'état e_i .



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

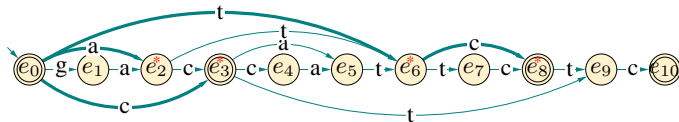
Conclusions &
perspectives

Définition (Facteurs Remarquables)

Étant donné un mot $s \in \Sigma^*$ et son Oracle, l'ensemble des facteurs remarquables de s , noté \mathcal{F}_s , est défini par :

$$\mathcal{F}_s = \{ \min(e_i) \mid 1 \leq i < |s| \wedge (\#_{out}(e_i) > 1 \vee \min(e_i) \in Suff(s)) \},$$

où $\#_{out}(e_i)$ désigne le degré sortant de l'état e_i .



$$\mathcal{F}_{gaccattctc} = \{a, c, t, tc\}$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

Conclusions &
perspectives

Exemple (contraction)

$$\mathcal{F}_{gaccattctc} = \{a, c, t, tc\}.$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

*Conclusions &
perspectives*

Exemple (contraction)

$$\mathcal{F}_{gaccattctc} = \{a, \mathbf{c}, t, tc\}.$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

*Conclusions &
perspectives*

Exemple (contraction)

$$\mathcal{F}_{gaccattctc} = \{a, \mathbf{c}, t, tc\}.$$

La paire (3, 8) est une contraction

$$\Rightarrow ga\mathbf{ccatt}ctc = gactc.$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

*Conclusions &
perspectives*

Exemple (Ensemble de contraction)

$$\mathcal{F}_{gaccattctc} = \{a, c, t, tc\} \Rightarrow \left\{ \begin{array}{l} (2, 5), (3, 4), \\ (3, 8), (3, 10), \\ (6, 7), (6, 9), \\ (7, 9) \end{array} \right\}$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

*Conclusions &
perspectives*

Exemple (Ensemble de contraction)

$$\mathcal{F}_{gaccattctc} = \{a, c, t, tc\} \Rightarrow \left\{ \begin{array}{l} (2, 5), (3, 4), \\ (3, 8), (3, 10), \\ (6, 7), (6, 9), \\ (7, 9) \end{array} \right\}$$

Exemple (Calcul des ensembles de contractions)

$$\left\{ \begin{array}{l} \{\} \\ \{(2, 5)\} \\ \{(3, 4)\} \\ \{(3, 8)\} \\ \{(3, 10)\} \\ \{(6, 7)\} \\ \{(6, 9)\} \\ \{(7, 9)\} \\ \{(2, 5), (6, 7)\} \\ \{(2, 5), (6, 9)\} \\ \{(2, 5), (7, 9)\} \\ \{(3, 4), (6, 7)\} \\ \{(3, 4), (6, 9)\} \\ \{(3, 4), (7, 9)\} \end{array} \right.$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

*Conclusions &
perspectives*

Exemple (Ensemble de contraction)

$$\mathcal{F}_{gaccattctc} = \{a, c, t, tc\} \Rightarrow \left\{ \begin{array}{l} (2, 5), (3, 4), \\ (3, 8), (3, 10), \\ (6, 7), (6, 9), \\ (7, 9) \end{array} \right\}$$

Exemple (Calcul des ensembles de contractions)

{}	\Rightarrow	<i>gaccattctc</i>
{(2, 5)}	\Rightarrow	<i>gaccattctc</i>
{(3, 4)}	\Rightarrow	<i>gaccattctc</i>
{(3, 8)}	\Rightarrow	<i>gaccattctc</i>
{(3, 10)}	\Rightarrow	<i>gaccattctc</i>
{(6, 7)}	\Rightarrow	<i>gaccattctc</i>
{(6, 9)}	\Rightarrow	<i>gaccattctc</i>
{(7, 9)}	\Rightarrow	<i>gaccattctc</i>
{(2, 5), (6, 7)}	\Rightarrow	<i>gaccattctc</i>
{(2, 5), (6, 9)}	\Rightarrow	<i>gaccattctc</i>
{(2, 5), (7, 9)}	\Rightarrow	<i>gaccattctc</i>
{(3, 4), (6, 7)}	\Rightarrow	<i>gaccattctc</i>
{(3, 4), (6, 9)}	\Rightarrow	<i>gaccattctc</i>
{(3, 4), (7, 9)}	\Rightarrow	<i>gaccattctc</i>

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

*Conclusions &
perspectives*

Exemple (Ensemble de contraction)

$$\mathcal{F}_{gaccattctc} = \{a, c, t, tc\} \Rightarrow \left\{ \begin{array}{l} (2, 5), (3, 4), \\ (3, 8), (3, 10), \\ (6, 7), (6, 9), \\ (7, 9) \end{array} \right\}$$

Exemple (Calcul des ensembles de contractions)

{ }	\Rightarrow	<i>gaccattctc</i>	=	<i>gaccattctc</i>
{(2, 5)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gattctc</i>
{(3, 4)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gacattctc</i>
{(3, 8)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gactc</i>
{(3, 10)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gac</i>
{(6, 7)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gaccatctc</i>
{(6, 9)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gaccatc</i>
{(7, 9)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gaccattc</i>
{(2, 5), (6, 7)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gatctc</i>
{(2, 5), (6, 9)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gatc</i>
{(2, 5), (7, 9)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gattc</i>
{(3, 4), (6, 7)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gacatctc</i>
{(3, 4), (6, 9)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gacatc</i>
{(3, 4), (7, 9)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gacattc</i>

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

Conclusions &
perspectives

Exemple (Fermeture)

La fermeture du mot *gaccattctc* est :

Exemple (Calcul des ensembles de contractions)

{}	⇒	<i>gaccattctc</i>	=	<i>gaccattctc</i>
{(2, 5)}	⇒	<i>gaccattctc</i>	=	<i>gattctc</i>
{(3, 4)}	⇒	<i>gaccattctc</i>	=	<i>gacattctc</i>
{(3, 8)}	⇒	<i>gaccattctc</i>	=	<i>gactc</i>
{(3, 10)}	⇒	<i>gaccattctc</i>	=	<i>gac</i>
{(6, 7)}	⇒	<i>gaccattctc</i>	=	<i>gaccatctc</i>
{(6, 9)}	⇒	<i>gaccattctc</i>	=	<i>gaccatc</i>
{(7, 9)}	⇒	<i>gaccattctc</i>	=	<i>gaccattc</i>
{(2, 5), (6, 7)}	⇒	<i>gaccattctc</i>	=	<i>gatctc</i>
{(2, 5), (6, 9)}	⇒	<i>gaccattctc</i>	=	<i>gatc</i>
{(2, 5), (7, 9)}	⇒	<i>gaccattctc</i>	=	<i>gattc</i>
{(3, 4), (6, 7)}	⇒	<i>gaccattctc</i>	=	<i>gacatctc</i>
{(3, 4), (6, 9)}	⇒	<i>gaccattctc</i>	=	<i>gacatc</i>
{(3, 4), (7, 9)}	⇒	<i>gaccattctc</i>	=	<i>gacattc</i>

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

Conclusions &
perspectives

Exemple (Fermeture)

La fermeture du mot *gaccattctc* est :

$$\mathcal{E}(gaccattctc) = \left\{ \begin{array}{l} gac, gacatc, gacatctc, gacattc, gacattctc, \\ gaccatc, gaccatctc, gaccattc, gaccattctc, \\ gactc, gatc, gatctc, gattc, gattctc \end{array} \right\}.$$

Exemple (Calcul des ensembles de contractions)

{}	\Rightarrow	<i>gaccattctc</i>	=	<i>gaccattctc</i>
{(2, 5)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gattctc</i>
{(3, 4)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gacattctc</i>
{(3, 8)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gactc</i>
{(3, 10)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gac</i>
{(6, 7)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gaccatctc</i>
{(6, 9)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gaccatc</i>
{(7, 9)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gaccattc</i>
{(2, 5), (6, 7)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gatctc</i>
{(2, 5), (6, 9)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gatc</i>
{(2, 5), (7, 9)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gattc</i>
{(3, 4), (6, 7)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gacatctc</i>
{(3, 4), (6, 9)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gacatc</i>
{(3, 4), (7, 9)}	\Rightarrow	<i>gaccattctc</i>	=	<i>gacattc</i>

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

Conclusions &
perspectives

Théorème (Langage reconnu par les Oracles des Suffixes)

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

*Conclusions &
perspectives*

Théorème (Langage reconnu par les Oracles des Suffixes)

L'Oracle des Suffixes de s reconnaît tous (et seulement) les suffixes des mots de $\mathcal{E}(s)$.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

*Conclusions &
perspectives*

Théorème (Langage reconnu par les Oracles des Suffixes)

L'Oracle des Suffixes de s reconnaît tous (et seulement) les suffixes des mots de $\mathcal{E}(s)$.

Exemple

Pour *gaccattctc*,

$$\mathcal{E}(gaccattctc) = \left\{ \begin{array}{l} gac, gacatc, gacatctc, gacattc, gacattctc, \\ gaccatc, gaccatctc, gaccattc, gaccattctc, \\ gactc, gatc, gatctc, gattc, gattctc \end{array} \right\}.$$

Il y a 43 mots reconnus par l'Oracle des Suffixes de *gaccattctc*, dont ses 11 suffixes.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

*Conclusions &
perspectives*

Théorème (Langage reconnu par les Oracles des Suffixes)

L'Oracle des Suffixes de s reconnaît tous (et seulement) les suffixes des mots de $\mathcal{E}(s)$.

Exemple

Pour *gaccattctc*,

$$\mathcal{E}(gaccattctc) = \left\{ \begin{array}{l} gac, gacatc, gacatctc, gacattc, gacattctc, \\ gaccatc, gaccatctc, gaccattc, gaccattctc, \\ gactc, gatc, gatctc, gattc, gattctc \end{array} \right\}.$$

Il y a 43 mots reconnus par l'Oracle des Suffixes de *gaccattctc*, dont ses 11 suffixes.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

*Conclusions &
perspectives*

Théorème (Langage reconnu par les Oracles des Suffixes)

L'Oracle des Suffixes de s reconnaît tous (et seulement) les suffixes des mots de $\mathcal{E}(s)$.

Exemple

Pour *gaccattctc*,

$$\mathcal{E}(gaccattctc) = \left\{ \begin{array}{l} gac, gacatc, gacatctc, gacattc, gacattctc, \\ gaccatc, gaccatctc, gaccattc, gaccattctc, \\ gactc, gatc, gatctc, gattc, gattctc \end{array} \right\}.$$

Il y a 43 mots reconnus par l'Oracle des Suffixes de *gaccattctc*, dont ses 11 suffixes.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

*Conclusions &
perspectives*

Théorème (Langage reconnu par les Oracles des Suffixes)

L'Oracle des Suffixes de s reconnaît tous (et seulement) les suffixes des mots de $\mathcal{E}(s)$.

Exemple

Pour *gaccattctc*,

$$\mathcal{E}(gaccattctc) = \left\{ \begin{array}{l} gac, gacatc, gacatctc, gacattc, gacattctc, \\ gaccatc, gaccatctc, gaccattc, gaccattctc, \\ gactc, gatc, gatctc, gattc, gattctc \end{array} \right\}.$$

Il y a 94 mots reconnus par l'Oracle des Facteurs de *gaccattctc*, dont ses 49 facteurs.

Théorème (Langage reconnu par les Oracles des Facteurs)

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

Conclusions &
perspectives

Théorème (Langage reconnu par les Oracles des Suffixes)

L'Oracle des Suffixes de s reconnaît tous (et seulement) les suffixes des mots de $\mathcal{E}(s)$.

Exemple

Pour *gaccattctc*,

$$\mathcal{E}(gaccattctc) = \left\{ \begin{array}{l} gac, gacatc, gacatctc, gacattc, gacattctc, \\ gaccatc, gaccatctc, gaccattc, gaccattctc, \\ gactc, gatc, gatctc, gattc, gattctc \end{array} \right\}.$$

Il y a 94 mots reconnus par l'Oracle des Facteurs de *gaccattctc*, dont ses 49 facteurs.

Théorème (Langage reconnu par les Oracles des Facteurs)

L'Oracle des Facteurs de s reconnaît tous (et seulement) les facteurs des mots de $\mathcal{E}(s)$.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

Conclusions &
perspectives

Théorème (Langage reconnu par les Oracles des Suffixes)

L'Oracle des Suffixes de s reconnaît tous (et seulement) les suffixes des mots de $\mathcal{E}(s)$.

Exemple

Pour *gaccattctc*,

$$\mathcal{E}(gaccattctc) = \left\{ \begin{array}{l} gac, gacatc, gacatctc, gacattc, gacattctc, \\ gaccatc, gaccatctc, gaccattc, gaccattctc, \\ gactc, gatc, gatctc, gattc, gattctc \end{array} \right\}.$$

Il y a 94 mots reconnus par l'Oracle des Facteurs de *gaccattctc*, dont ses 49 facteurs.

Théorème (Langage reconnu par les Oracles des Facteurs)

L'Oracle des Facteurs de s reconnaît tous (et seulement) les facteurs des mots de $\mathcal{E}(s)$.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Facteurs remarquables

Opération de contraction & fermeture

Caractérisation du langage

Vers un nouvel Oracle

Conclusions &
perspectives

- Un mot rejeté par l'Oracle des Facteurs/Suffixes d'une séquence s n'est pas facteur/suffixe de s ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- Un mot rejeté par l'Oracle des Facteurs/Suffixes d'une séquence s n'est pas facteur/suffixe de s ;
- Un mot accepté par l'Oracle des Facteurs/Suffixes d'une séquence s n'est pas toujours facteur/suffixe de s .

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- Un mot rejeté par l'Oracle des Facteurs/Suffixes d'une séquence s n'est pas facteur/suffixe de s ;
- Un mot accepté par l'Oracle des Facteurs/Suffixes d'une séquence s n'est pas toujours facteur/suffixe de s .
 - Notion de Vrai/Faux Positifs ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- Un mot rejeté par l'Oracle des Facteurs/Suffixes d'une séquence s n'est pas facteur/suffixe de s ;
- Un mot accepté par l'Oracle des Facteurs/Suffixes d'une séquence s n'est pas toujours facteur/suffixe de s .
 - Notion de Vrai/Faux Positifs ;
- Nombre de « Faux Positifs » peut-être exponentiel par rapport à $|s|$;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- Un mot rejeté par l'Oracle des Facteurs/Suffixes d'une séquence s n'est pas facteur/suffixe de s ;
- Un mot accepté par l'Oracle des Facteurs/Suffixes d'une séquence s n'est pas toujours facteur/suffixe de s .
 - Notion de Vrai/Faux Positifs ;
- Nombre de « Faux Positifs » peut-être exponentiel par rapport à $|s|$;
 - Faux Positifs engendrés par « contractions » (\Rightarrow arcs externes).

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- **Objectif :**

- gardes sur les transitions externes des Oracles.

- **Calcul :**

- mise à jour à partir de l'Oracle de $s[1..i]$.

- **Utilisation :**

- filtrer le passage par une transition externe selon la garde présente.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- **Objectif :**

- gardes sur les transitions externes des Oracles.

- **Calcul :**

- mise à jour à partir de l'Oracle de $s[1..i]$.

- **Utilisation :**

- filtrer le passage par une transition externe selon la garde présente.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idée principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- **Objectif :**

- gardes sur les transitions externes des Oracles.

- **Calcul :**

- mise à jour à partir de l'Oracle de $s[1..i]$.

- **Utilisation :**

- filtrer le passage par une transition externe selon la garde présente.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- **Objectif :**
 - gardes sur les transitions externes des Oracles.
- **Calcul :**
 - mise à jour à partir de l'Oracle de $s[1..i]$.
- **Utilisation :**
 - filtrer le passage par une transition externe selon la garde présente.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idée principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .

Suffixes de s	Classes	État de reconnaissance
-----	Classe 1	$e_{ s }$

Oracle à transitions gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

Conclusions & perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .



Oracle à transitions gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

Conclusions & perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .



Oracle à transitions gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

Conclusions & perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .



Oracle à transitions gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

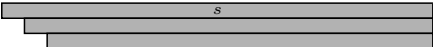
Critères d'évaluation

Performances respectives

Conclusions & perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .

Suffixes de s	Classes	État de reconnaissance
	Classe 1	$e_{ s }$

Oracle à transitions gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

Un second exemple

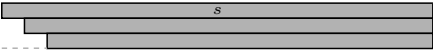
Critères d'évaluation

Performances respectives

Conclusions & perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .

Suffixes de s	Classes	État de reconnaissance
	Classe I	$e_{ s }$
	Classe II	$S_s(e_{ s })$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .

Suffixes de s	Classes	État de reconnaissance
	Classe I	$e_{ s }$
	Classe II	$S_s(e_{ s })$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .

Suffixes de s	Classes	État de reconnaissance
	Classe I	$e_{ s }$
	Classe II	$S_s(e_{ s })$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

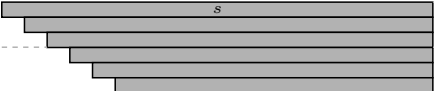
Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .

Suffixes de s	Classes	État de reconnaissance
	Classe I	$e_{ s }$
	Classe II	$S_s(e_{ s })$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

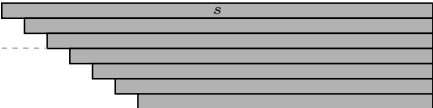
Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .

Suffixes de s	Classes	État de reconnaissance
	Classe I	$e_{ s }$
	Classe II	$S_s(e_{ s })$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

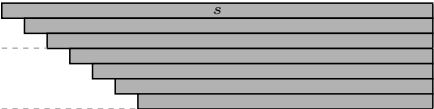
Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .

Suffixes de s	Classes	État de reconnaissance
	Classe I	$e_{ s }$
	Classe II	$S_s(e_{ s })$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .

Suffixes de s	Classes	État de reconnaissance
	Classe I	$e_{ s }$
	Classe II	$S_s(e_{ s })$
	Classe III	$S_s^2(e_{ s })$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .

Suffixes de s	Classes	État de reconnaissance
	Classe I	$e_{ s }$
	Classe II	$S_s(e_{ s })$
	Classe III	$S_s^2(e_{ s })$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

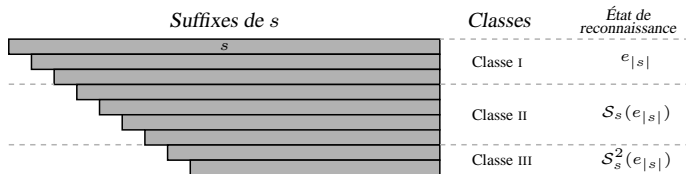
Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .



Oracle à transitions gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

Conclusions & perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .

Suffixes de s	Classes	État de reconnaissance
	Classe I	$e_{ s }$
	Classe II	$S_s(e_{ s })$
	Classe III	$S_s^2(e_{ s })$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

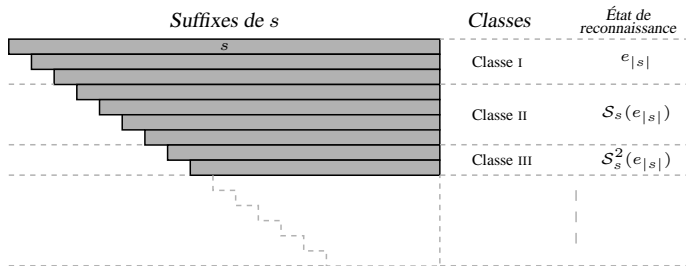
Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

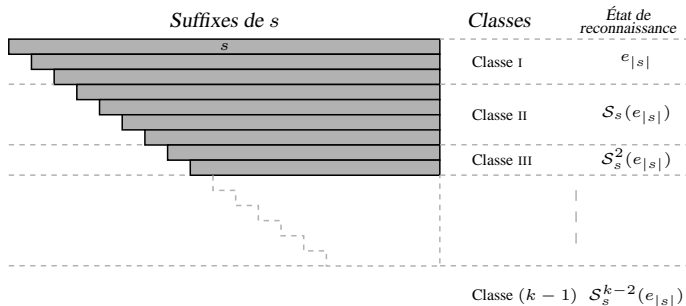
Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .



Oracle à transitions gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

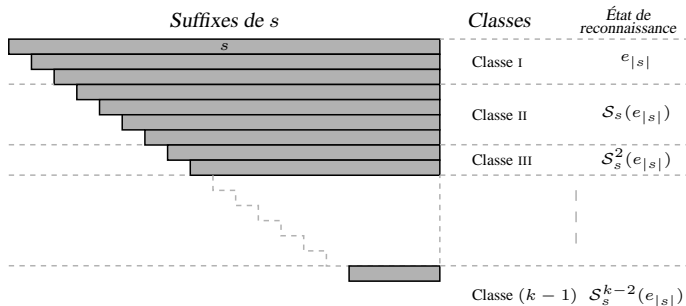
Critères d'évaluation

Performances respectives

Conclusions & perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

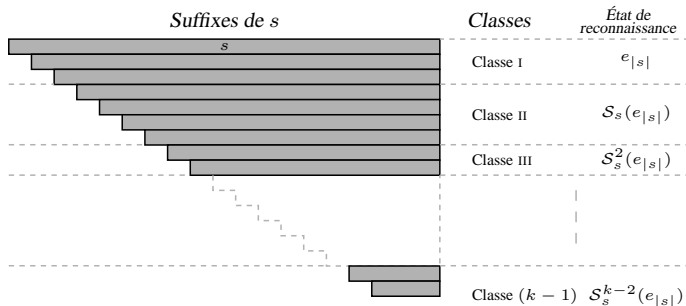
Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

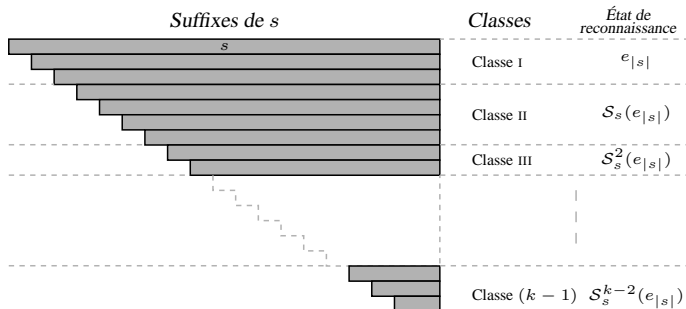
Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

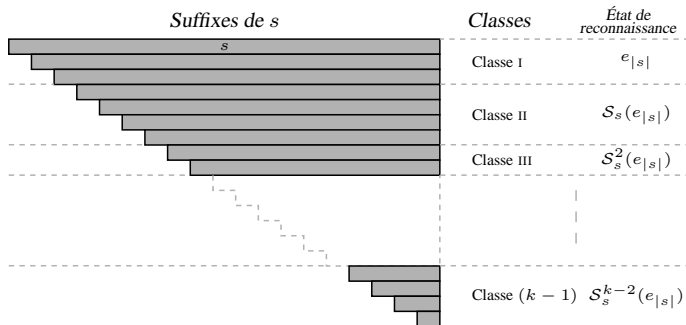
Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .



Oracle à transitions gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

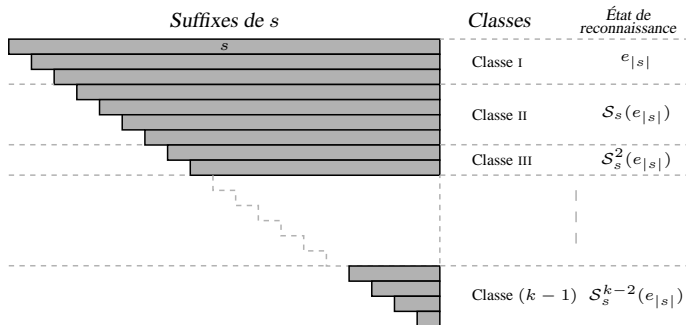
Critères d'évaluation

Performances respectives

Conclusions & perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .



Oracle à transitions gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

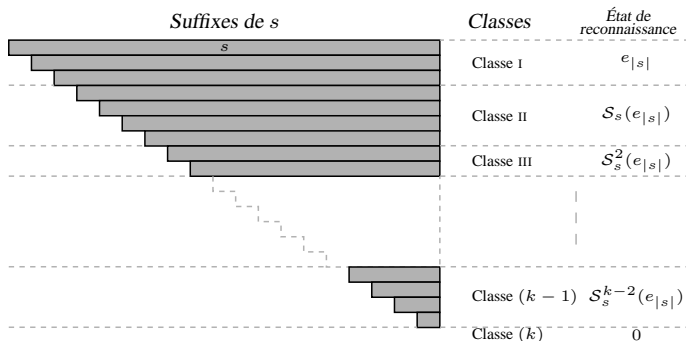
Critères d'évaluation

Performances respectives

Conclusions & perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .



Oracle à transitions gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

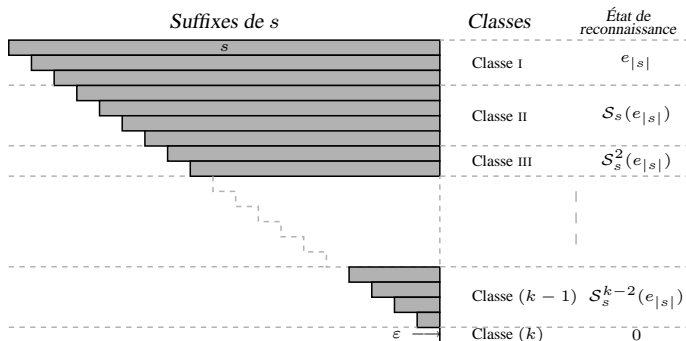
Critères d'évaluation

Performances respectives

Conclusions & perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .



Oracle à transitions gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

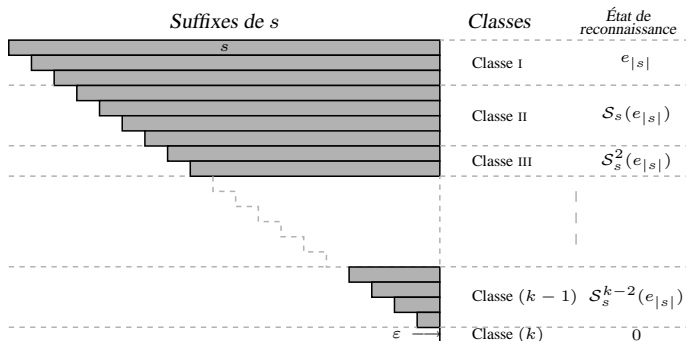
Critères d'évaluation

Performances respectives

Conclusions & perspectives

Définition (Classes de suffixes)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, l'ensemble $Suff(s)$ peut être partitionné de manière à ce que tous les suffixes d'une même partie soient reconnus dans le même état de l'Oracle de s . Chaque partie est alors une classe de suffixes de s .



Oracle à transitions gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

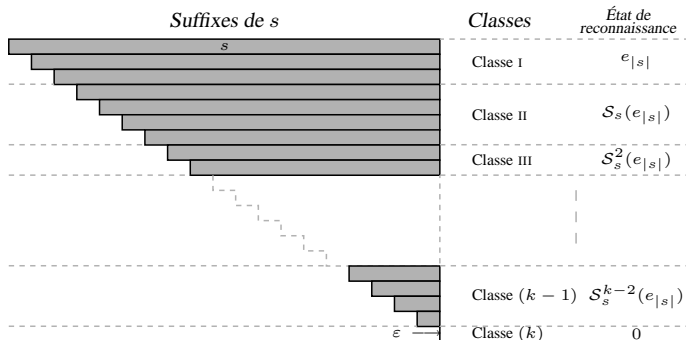
Critères d'évaluation

Performances respectives

Conclusions & perspectives

Fonction de suppléance sur les états : \mathcal{S}_s

⇒ identification de l'état de reconnaissance du plus grand suffixe répété de $s[1..i]$.



Oracle à transitions gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

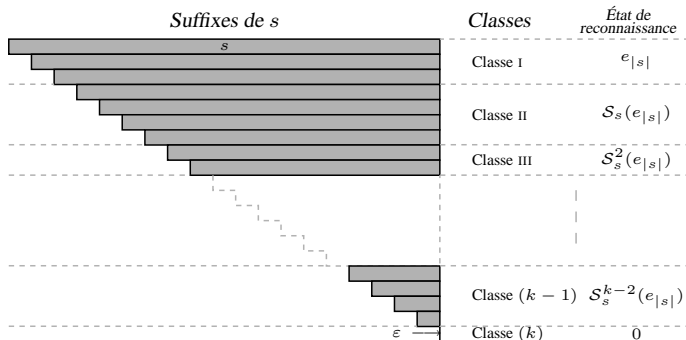
Critères d'évaluation

Performances respectives

Conclusions & perspectives

Fonction de suppléance sur les états : \mathcal{S}_s

~~⇒ identification de l'état de reconnaissance du plus grand suffixe répété de $s[1..i]$.~~



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

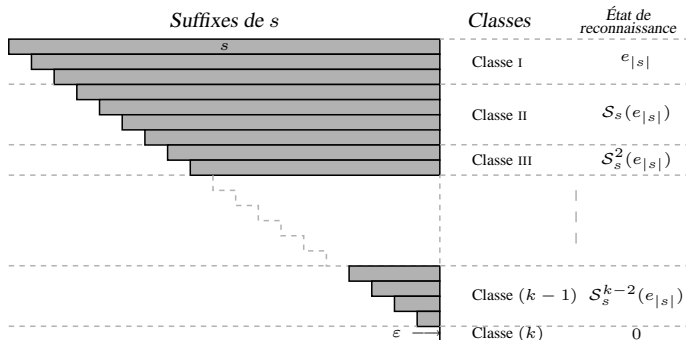
Performances respectives

Conclusions &
perspectives

Fonction de suppléance sur les états : \mathcal{S}_s

~~⇒ identification de l'état de reconnaissance du plus grand suffixe répété de $s[1..i]$.~~

⇒ identification de l'état de reconnaissance des suffixes de $s[1..i]$ de la classe suivante.



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Définition (Fonction succédanée)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, la fonction succédanée \mathcal{L}_s est définie comme étant la fonction qui identifie, pour chaque état e_i ($i \leq 1 \leq |s|$) de l'Oracle de s , la longueur du plus grand suffixe de $s[1..i]$ reconnu à l'état $\mathcal{S}_s(e_i)$.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Définition (Fonction succédanée)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, la fonction succédanée \mathcal{L}_s est définie comme étant la fonction qui identifie, pour chaque état e_i ($i \leq 1 \leq |s|$) de l'Oracle de s , la longueur du plus grand suffixe de $s[1..i]$ reconnu à l'état $\mathcal{S}_s(e_i)$.

- Principe de la m-à-j. à partir de l'Oracle de $s[1..i]$:
 - ajout de l'état e_{i+1} ;
 - ajout de la transition entre e_i et e_{i+1} étiqueté par $s[i + 1]$;
 - ajout des transitions externes *gardées* arrivant en e_{i+1} .
- Calcul de la valeur de garde :

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idée principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Définition (Fonction succédanée)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, la fonction succédanée \mathcal{L}_s est définie comme étant la fonction qui identifie, pour chaque état e_i ($i \leq 1 \leq |s|$) de l'Oracle de s , la longueur du plus grand suffixe de $s[1..i]$ reconnu à l'état $\mathcal{S}_s(e_i)$.

- Principe de la m-à-j. à partir de l'Oracle de $s[1..i]$:
 - ajout de l'état e_{i+1} ;
 - ajout de la transition entre e_i et e_{i+1} étiqueté par $s[i + 1]$;
 - ajout des transitions externes *gardées* arrivant en e_{i+1} .
- Calcul de la valeur de garde :

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Définition (Fonction succédanée)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, la fonction succédanée \mathcal{L}_s est définie comme étant la fonction qui identifie, pour chaque état e_i ($i \leq 1 \leq |s|$) de l'Oracle de s , la longueur du plus grand suffixe de $s[1..i]$ reconnu à l'état $\mathcal{S}_s(e_i)$.

- Principe de la m-à-j. à partir de l'Oracle de $s[1..i]$:
 - ajout de l'état e_{i+1} ;
 - ajout de la transition entre e_i et e_{i+1} étiqueté par $s[i + 1]$;
 - ajout des transitions externes *gardées* arrivant en e_{i+1} .
- Calcul de la valeur de garde :

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idée principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Définition (Fonction succédanée)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, la fonction succédanée \mathcal{L}_s est définie comme étant la fonction qui identifie, pour chaque état e_i ($i \leq 1 \leq |s|$) de l'Oracle de s , la longueur du plus grand suffixe de $s[1..i]$ reconnu à l'état $\mathcal{S}_s(e_i)$.

- Principe de la m-à-j. à partir de l'Oracle de $s[1..i]$:
 - ajout de l'état e_{i+1} ;
 - ajout de la transition entre e_i et e_{i+1} étiqueté par $s[i + 1]$;
 - ajout des transitions externes *gardées* arrivant en e_{i+1} .
- Calcul de la valeur de garde :
 - utilisation des fonctions de suppléance et succédanée \mathcal{S}_s et \mathcal{L}_s ;
 - mise à jour de \mathcal{S}_s et de \mathcal{L}_s .

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Définition (Fonction succédanée)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, la fonction succédanée \mathcal{L}_s est définie comme étant la fonction qui identifie, pour chaque état e_i ($i \leq 1 \leq |s|$) de l'Oracle de s , la longueur du plus grand suffixe de $s[1..i]$ reconnu à l'état $\mathcal{S}_s(e_i)$.

- Principe de la m-à-j. à partir de l'Oracle de $s[1..i]$:
 - ajout de l'état e_{i+1} ;
 - ajout de la transition entre e_i et e_{i+1} étiqueté par $s[i + 1]$;
 - ajout des transitions externes *gardées* arrivant en e_{i+1} .
- Calcul de la valeur de garde :
 - utilisation des fonctions de suppléance et succédanée \mathcal{S}_s et \mathcal{L}_s ;
 - mise à jour de \mathcal{S}_s et de \mathcal{L}_s .

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idée principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Définition (Fonction succédanée)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, la fonction succédanée \mathcal{L}_s est définie comme étant la fonction qui identifie, pour chaque état e_i ($i \leq 1 \leq |s|$) de l'Oracle de s , la longueur du plus grand suffixe de $s[1..i]$ reconnu à l'état $\mathcal{S}_s(e_i)$.

- Principe de la m-à-j. à partir de l'Oracle de $s[1..i]$:
 - ajout de l'état e_{i+1} ;
 - ajout de la transition entre e_i et e_{i+1} étiqueté par $s[i + 1]$;
 - ajout des transitions externes *gardées* arrivant en e_{i+1} .
- Calcul de la valeur de garde :
 - utilisation des fonctions de suppléance et succédanée \mathcal{S}_s et \mathcal{L}_s ;
 - mise à jour de \mathcal{S}_s et de \mathcal{L}_s .

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idée principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Définition (Fonction succédanée)

Étant donné un mot $s \in \Sigma^+$ et son Oracle, la fonction succédanée \mathcal{L}_s est définie comme étant la fonction qui identifie, pour chaque état e_i ($i \leq 1 \leq |s|$) de l'Oracle de s , la longueur du plus grand suffixe de $s[1..i]$ reconnu à l'état $\mathcal{S}_s(e_i)$.

- Principe de la m-à-j. à partir de l'Oracle de $s[1..i]$:
 - ajout de l'état e_{i+1} ;
 - ajout de la transition entre e_i et e_{i+1} étiqueté par $s[i + 1]$;
 - ajout des transitions externes *gardées* arrivant en e_{i+1} .
- Calcul de la valeur de garde :
 - utilisation des fonctions de suppléance et succédanée \mathcal{S}_s et \mathcal{L}_s ;
 - mise à jour de \mathcal{S}_s et de \mathcal{L}_s .

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idée principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Vers un nouvel oracle

Nouvel algorithme « on-line »

```
1 Entrées :  $\Sigma$  % Alphabet (supposé minimal) %
2            $s \in \Sigma^*$  % Le mot à traiter %
3 Sortie : Oracle % L'Oracle à transitions gardées de  $s$  %
4 Début
5   Créer l'état initial étiqueté par  $e_0$ 
6    $\mathcal{S}_s(e_0) \leftarrow e_{-1}$ 
7    $\mathcal{L}_s(e_0) \leftarrow -1$ 
8   Pour  $i$  de 1 à  $|s|$  Faire
9     Créer un état étiqueté par  $e_i$ 
10    Ajouter une transition de l'état  $e_{i-1}$  vers l'état  $e_i$  étiquetée par  $s[i]$ 
11     $l \leftarrow \mathcal{L}_s(e_{i-1})$  % Longueur du plus long suffixe répété de  $s[1..i-1]$  %
12     $e_k \leftarrow \mathcal{S}_s(e_{i-1})$ 
13    Tant Que  $k > -1$  et
14      qu'il n'existe pas de transition issue de  $e_k$  étiquetée par  $s[i]$  Faire
15      Ajouter une transition de l'état  $e_k$  vers l'état  $e_i$  étiquetée par  $s[i]$ 
16      Poser  $l$  comme valeur de garde pour cette transition.
17       $l \leftarrow \mathcal{L}_s(e_k)$  % Longueur du plus long suffixe répété de  $s[1..k]$  %
18       $e_k \leftarrow \mathcal{S}_s(e_k)$ 
19    Fin Tant Que
20    Si ( $k = -1$ ) Alors
21       $\mathcal{S}_s(e_i) \leftarrow e_0$ 
22       $\mathcal{L}_s(e_i) \leftarrow 0$ 
23    Sinon
24      Soit  $e_x$  l'état d'arrivée de la transition issue de  $e_k$  étiquetée par  $s[i]$ 
25       $\mathcal{S}_s(e_i) \leftarrow e_x$ 
26       $\mathcal{L}_s(e_i) \leftarrow l + 1$ 
27      Si la transition issue de  $e_k$  vers  $e_x$  par  $s[i+1]$  est externe Alors
28         $l' \leftarrow$  valeur actuelle de la garde de cette transition.
29        Poser  $\max(l, l')$  comme nouvelle valeur de garde pour cette transition.
30      Fin Si
31    Fin Si
32  Fin Pour
33 Fin
```

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 1 \quad s[i + 1] = g$$

$$\mathcal{S}_\varepsilon(0) = -1$$

$$\mathcal{L}_\varepsilon(0) = -1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

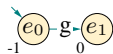
Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 1 \quad s[i + 1] = g$$

$$\mathcal{S}_\varepsilon(0) = -1$$

$$\mathcal{L}_\varepsilon(0) = -1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

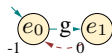
Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 2 \quad s[i + 1] = a$$

$$\mathcal{S}_g(1) = 0$$

$$\mathcal{L}_g(1) = 0$$

$$\mathcal{S}_g(0) = -1$$

$$\mathcal{L}_g(0) = -1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

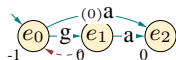
Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 2 \quad s[i + 1] = a$$

$$\mathcal{S}_g(1) = 0$$

$$\mathcal{L}_g(1) = 0$$

$$\mathcal{S}_g(0) = -1$$

$$\mathcal{L}_g(0) = -1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

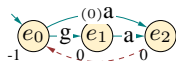
Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 3 \quad s[i + 1] = c$$

$$\mathcal{S}_{ga}(2) = 0$$

$$\mathcal{L}_{ga}(2) = 0$$

$$\mathcal{S}_{ga}(0) = -1$$

$$\mathcal{L}_{ga}(0) = -1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

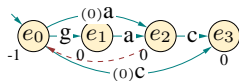
Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 3 \quad s[i + 1] = c$$

$$\mathcal{S}_{ga}(2) = 0$$

$$\mathcal{L}_{ga}(2) = 0$$

$$\mathcal{S}_{ga}(0) = -1$$

$$\mathcal{L}_{ga}(0) = -1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

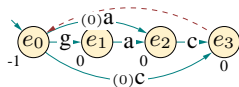
Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 4 \quad s[i + 1] = c$$

$$\mathcal{S}_{gac}(3) = 0$$

$$\mathcal{S}_{gac}(0) = -1$$

$$\mathcal{L}_{gac}(0) = 0$$

$$\mathcal{L}_{gac}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

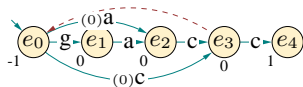
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 4 \quad s[i + 1] = c$$

$$\mathcal{S}_{gac}(3) = 0$$

$$\mathcal{S}_{gac}(0) = -1$$

$$\mathcal{L}_{gac}(0) = 0$$

$$\mathcal{L}_{gac}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

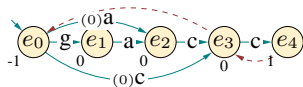
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 5 \quad s[i + 1] = a$$

$$\mathcal{S}_{gacc}(4) = 3$$

$$\mathcal{L}_{gacc}(4) = 1$$

$$\mathcal{S}_{gacc}(3) = 0$$

$$\mathcal{L}_{gacc}(3) = 0$$

$$\mathcal{S}_{gacc}(0) = -1$$

$$\mathcal{L}_{gacc}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

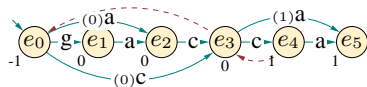
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 5 \quad s[i + 1] = a$$

$$S_{gacc}(4) = 3$$

$$\mathcal{L}_{gacc}(4) = 1$$

$$S_{gacc}(3) = 0$$

$$\mathcal{L}_{gacc}(3) = 0$$

$$S_{gacc}(0) = -1$$

$$\mathcal{L}_{gacc}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

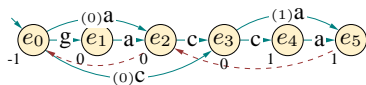
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 6 \quad s[i + 1] = t$$

$$S_{gacca}(5) = 2$$

$$\mathcal{L}_{gacca}(5) = 1$$

$$S_{gacca}(2) = 0$$

$$\mathcal{L}_{gacca}(2) = 0$$

$$S_{gacca}(0) = -1$$

$$\mathcal{L}_{gacca}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

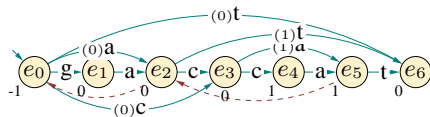
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 6 \quad s[i + 1] = t$$

$$S_{gacca}(5) = 2$$

$$\mathcal{L}_{gacca}(5) = 1$$

$$S_{gacca}(2) = 0$$

$$\mathcal{L}_{gacca}(2) = 0$$

$$S_{gacca}(0) = -1$$

$$\mathcal{L}_{gacca}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

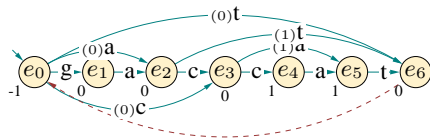
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 7 \quad s[i + 1] = t$$

$$\mathcal{S}_{gaccat}(6) = 0$$

$$\mathcal{L}_{gaccat}(6) = 0$$

$$\mathcal{S}_{gaccat}(0) = -1$$

$$\mathcal{L}_{gaccat}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idée principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

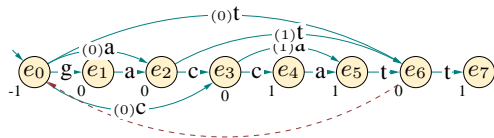
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 7 \quad s[i + 1] = t$$

$$\mathcal{S}_{gaccat}(6) = 0$$

$$\mathcal{L}_{gaccat}(6) = 0$$

$$\mathcal{S}_{gaccat}(0) = -1$$

$$\mathcal{L}_{gaccat}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

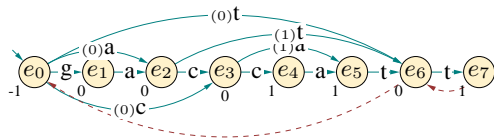
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 8 \quad s[i + 1] = c$$

$$S_{gaccatt}(7) = 6$$

$$\mathcal{L}_{gaccatt}(7) = 1$$

$$S_{gaccatt}(6) = 0$$

$$\mathcal{L}_{gaccatt}(6) = 0$$

$$S_{gaccatt}(0) = -1$$

$$\mathcal{L}_{gaccatt}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

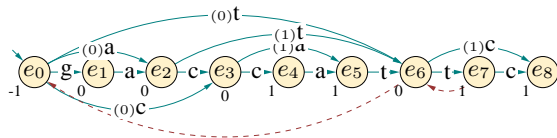
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 8 \quad s[i + 1] = c$$

$$S_{gaccatt}(7) = 6$$

$$\mathcal{L}_{gaccatt}(7) = 1$$

$$S_{gaccatt}(6) = 0$$

$$\mathcal{L}_{gaccatt}(6) = 0$$

$$S_{gaccatt}(0) = -1$$

$$\mathcal{L}_{gaccatt}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

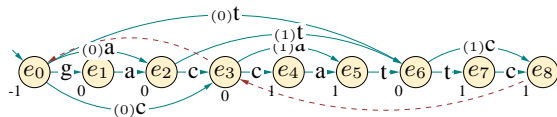
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 9 \quad s[i + 1] = t$$

$$S_{gaccattc}(8) = 3$$

$$\mathcal{L}_{gaccattc}(8) = 1$$

$$S_{gaccattc}(3) = 0$$

$$\mathcal{L}_{gaccattc}(3) = 0$$

$$S_{gaccattc}(0) = -1$$

$$\mathcal{L}_{gaccattc}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

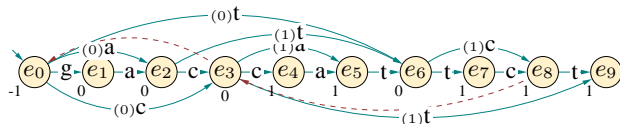
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 9 \quad s[i + 1] = t$$

$$S_{gaccattc}(8) = 3$$

$$\mathcal{L}_{gaccattc}(8) = 1$$

$$S_{gaccattc}(3) = 0$$

$$\mathcal{L}_{gaccattc}(3) = 0$$

$$S_{gaccattc}(0) = -1$$

$$\mathcal{L}_{gaccattc}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idée principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

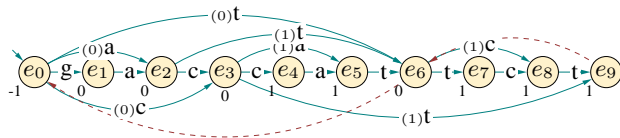
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 10 \quad s[i + 1] = c$$

$$S_{gaccattct}(9) = 3$$

$$\mathcal{L}_{gaccattct}(9) = 1$$

$$S_{gaccattct}(3) = 0$$

$$\mathcal{L}_{gaccattct}(3) = 0$$

$$S_{gaccattct}(0) = -1$$

$$\mathcal{L}_{gaccattct}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

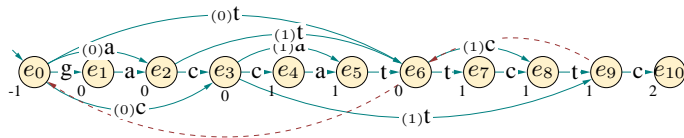
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$i + 1 = 10 \quad s[i + 1] = c$$

$$S_{gaccattct}(9) = 3$$

$$\mathcal{L}_{gaccattct}(9) = 1$$

$$S_{gaccattct}(3) = 0$$

$$\mathcal{L}_{gaccattct}(3) = 0$$

$$S_{gaccattct}(0) = -1$$

$$\mathcal{L}_{gaccattct}(0) = -1$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

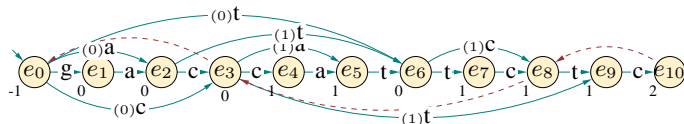
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



$$S_{gaccattctc}(10) = 8 \quad \mathcal{L}_{gaccattctc}(10) = 2$$

$$S_{gaccattctc}(8) = 3 \quad \mathcal{L}_{gaccattctc}(8) = 1$$

$$S_{gaccattctc}(3) = 0 \quad \mathcal{L}_{gaccattctc}(3) = 0$$

Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

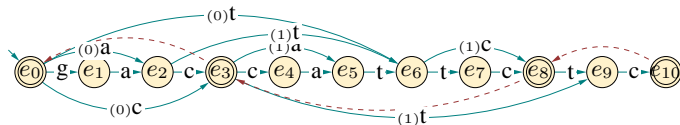
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

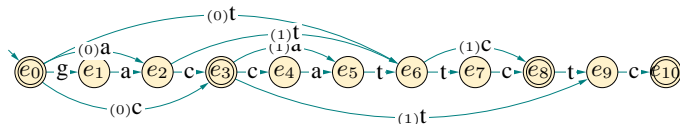
Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Oracle des Suffixes à transitions gardées du mot *gaccattctc*



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

Un second exemple

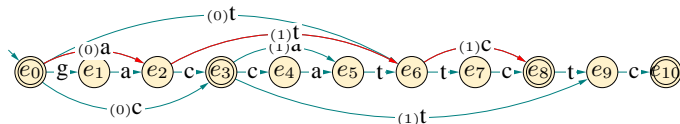
Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*

$atc \notin \text{Suff}(gaccattctc)$



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

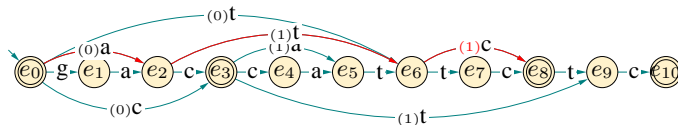
Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Suffixes à transitions gardées du mot *gaccattctc*

$atc \notin \text{Suff}(gaccattctc)$



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

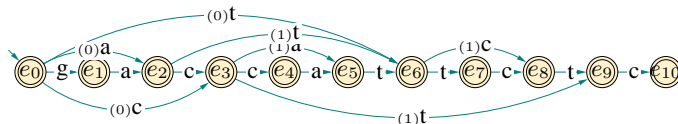
Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

Oracle des Facteurs à transitions gardées du mot *gaccattctc*



Oracle à transitions
gardées

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

Langage reconnu par les
Oracles

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

Conclusions &
perspectives

- Plusieurs paramètres :

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

● Plusieurs paramètres :

● Plusieurs critères :

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- Plusieurs paramètres :
 - taille de l'alphabet ($|\Sigma|$);
- Plusieurs critères :

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- Plusieurs paramètres :

- taille de l'alphabet ($|\Sigma|$);
- taille des séquences
($|s| = n$)...

- Plusieurs critères :

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- Plusieurs paramètres :

- taille de l'alphabet ($|\Sigma|$);
- taille des séquences
($|s| = n$)...

- Plusieurs critères :

- nombre de mots à reconnaître ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- Plusieurs paramètres :

- taille de l'alphabet ($|\Sigma|$);
- taille des séquences
($|s| = n$)...

- Plusieurs critères :

- nombre de mots à reconnaître ;
- nombre de mots reconnus ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- Plusieurs paramètres :

- taille de l'alphabet ($|\Sigma|$);
- taille des séquences
($|s| = n$)...

- Plusieurs critères :

- nombre de mots à reconnaître ;
- nombre de mots reconnus ;
- nombre de transitions externes...

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- Plusieurs paramètres :

- taille de l'alphabet ($|\Sigma|$);
- taille des séquences
($|s| = n$)...

- Plusieurs critères :

- nombre de mots à reconnaître ;
- nombre de mots reconnus ;
- nombre de transitions externes...

Principe des tests statistiques :

	Critère validé	Critère non validé
Test positif	Vrai Positifs (VP)	Faux Positifs (FP)
Test négatif	Faux Négatifs (FN)	Vrai Négatifs (VN)

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- Plusieurs paramètres :

- taille de l'alphabet ($|\Sigma|$);
- taille des séquences
($|s| = n$)...

- Plusieurs critères :

- nombre de mots à reconnaître ;
- nombre de mots reconnus ;
- nombre de transitions externes...

Principe des tests statistiques :

	Mots à accepter	Mots à rejeter
Mots acceptés	Vrai Positifs (VP)	Faux Positifs (FP)
Mots rejetés	Faux Négatifs (FN)	Vrai Négatifs (VN)

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédant en suppléant

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- Plusieurs paramètres :

- taille de l'alphabet ($|\Sigma|$);
- taille des séquences
($|s| = n$)...

- Plusieurs critères :

- nombre de mots à reconnaître ;
- nombre de mots reconnus ;
- nombre de transitions externes...

Principe des tests statistiques :

	Mots à accepter	Mots à rejeter
Mots acceptés	Vrai Positifs (VP)	Faux Positifs (FP)
Mots rejetés	Faux Négatifs (FN)=0	Vrai Négatifs (VN)

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

● Plusieurs paramètres :

- taille de l'alphabet ($|\Sigma|$);
- taille des séquences ($|s| = n$)...

● Plusieurs critères :

- nombre de mots à reconnaître ;
- nombre de mots reconnus ;
- nombre de transitions externes...

Principe des tests statistiques :

	Mots à accepter	Mots à rejeter
Mots acceptés	Vrai Positifs (VP)	Faux Positifs (FP)
Mots rejetés	Faux Négatifs (FN)=0	Vrai Négatifs (VN)

Intensité de liaison entre un test et la réalité \Rightarrow Coefficient de YULE :

$$Q = \frac{VP \times VN - FP \times FN}{VP \times VN + FP \times FN}$$

$Q \leq 0$	nulla,
$0 < Q < 0,1$	négligeable,
$0,1 \leq Q < 0,3$	légère,
$0,3 \leq Q < 0,5$	modérée,
$0,5 \leq Q < 0,7$	forte,
$0,7 \leq Q \leq 1$	très forte.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

● Plusieurs paramètres :

- taille de l'alphabet ($|\Sigma|$);
- taille des séquences ($|s| = n$)...

● Plusieurs critères :

- nombre de mots à reconnaître ;
- nombre de mots reconnus ;
- nombre de transitions externes...

Principe des tests statistiques :

	Mots à accepter	Mots à rejeter
Mots acceptés	Vrai Positifs (VP)	Faux Positifs (FP)
Mots rejetés	Faux Négatifs (FN)=0	Vrai Négatifs (VN)

Intensité de liaison entre un test et la réalité \Rightarrow Coefficient de YULE :

$$Q = \frac{VP \times VN - FP \times FN}{VP \times VN + FP \times FN} = 1$$

$Q \leq 0$	nulle,
$0 < Q < 0,1$	négligeable,
$0,1 \leq Q < 0,3$	légère,
$0,3 \leq Q < 0,5$	modérée,
$0,5 \leq Q < 0,7$	forte,
$0,7 \leq Q \leq 1$	très forte.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- Plusieurs paramètres :

- taille de l'alphabet ($|\Sigma|$);
- taille des séquences
($|s| = n$)...

- Plusieurs critères :

- nombre de mots à reconnaître ;
- nombre de mots reconnus ;
- nombre de transitions externes...

Principe des tests statistiques :

	Mots à accepter	Mots à rejeter
Mots acceptés	Vrai Positifs (VP)	Faux Positifs (FP)
Mots rejetés	Faux Négatifs (FN)=0	Vrai Négatifs (VN)

Indicateurs sur la fiabilité des tests statistiques :

$$\text{Sensibilité} = \frac{VP}{VP + FN}$$

$$\text{Spécificité} = \frac{VN}{VN + FP}$$

$$VPP = \frac{VP}{VP + FP}$$

$$VPN = \frac{VN}{VN + FN}$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

● Plusieurs paramètres :

- taille de l'alphabet ($|\Sigma|$);
- taille des séquences
($|s| = n$)...

● Plusieurs critères :

- nombre de mots à reconnaître ;
- nombre de mots reconnus ;
- nombre de transitions externes...

Principe des tests statistiques :

	Mots à accepter	Mots à rejeter
Mots acceptés	Vrai Positifs (VP)	Faux Positifs (FP)
Mots rejetés	Faux Négatifs (FN)=0	Vrai Négatifs (VN)

Indicateurs sur la fiabilité des tests statistiques :

$$\text{Sensibilité} = \frac{VP}{VP + FN} = 1$$

$$\text{Spécificité} = \frac{VN}{VN + FP}$$

$$VPP = \frac{VP}{VP + FP}$$

$$VPN = \frac{VN}{VN + FN} = 1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

● Plusieurs paramètres :

- taille de l'alphabet ($|\Sigma|$);
- taille des séquences
($|s| = n$)...

● Plusieurs critères :

- nombre de mots à reconnaître ;
- nombre de mots reconnus ;
- nombre de transitions externes...

Principe des tests statistiques :

	Mots à accepter	Mots à rejeter
Mots acceptés	Vrai Positifs (VP)	Faux Positifs (FP)
Mots rejetés	Faux Négatifs (FN)=0	Vrai Négatifs (VN)

Indicateurs sur la fiabilité des tests statistiques :

$$\text{Sensibilité} = \frac{\text{VP}}{\text{VP} + \text{FN}} = 1$$

$$\text{Spécificité} = \frac{\text{VN}}{\text{VN} + \text{FP}} \approx 1$$

$$\text{VPP} = \frac{\text{VP}}{\text{VP} + \text{FP}}$$

$$\text{VPN} = \frac{\text{VN}}{\text{VN} + \text{FN}} = 1$$

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

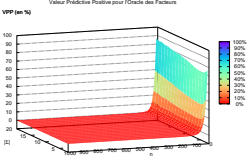
Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Vers un nouvel Oracle

Performances respectives [VPP]

Oracles	Facteurs	Suffixes
Version Originale		
Transitions Gardées		

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en supplém

Nouvel algorithme « on-line »

Un second exemple

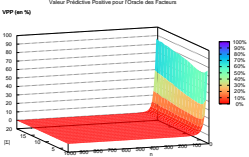
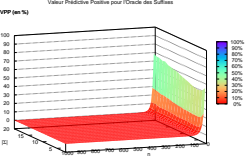
Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Vers un nouvel Oracle

Performances respectives [VPP]

Oracles	Facteurs	Suffixes
Version Originale	 <p>Valueur Prédicive Positive pour l'Oracle des Facteurs</p> <p>VPP (en %)</p> <p>100% 90% 80% 70% 60% 50% 40% 30% 20% 10% 0%</p>	 <p>Valueur Prédicive Positive pour l'Oracle des Suffixes</p> <p>VPP (en %)</p> <p>100% 90% 80% 70% 60% 50% 40% 30% 20% 10% 0%</p>
Transitions Gardées		

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

Un second exemple

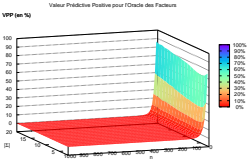
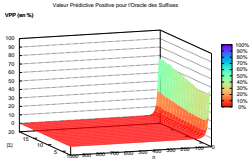
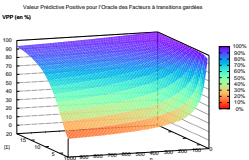
Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Vers un nouvel Oracle

Performances respectives [VPP]

Oracles	Facteurs	Suffixes
Version Originale	<p> Valeur Prédicative Positive pour l'Oracle des Facteurs</p> 	<p> Valeur Prédicative Positive pour l'Oracle des Suffixes</p> 
Transitions Gardées	<p> Valeur Prédicative Positive pour l'Oracle des Facteurs à transitions gardées</p> 	

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

Un second exemple

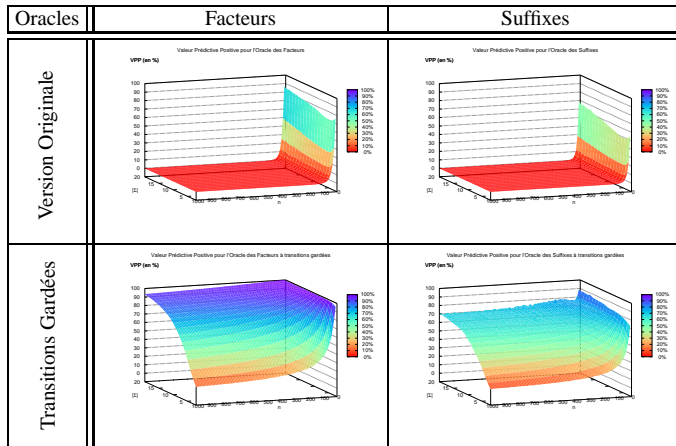
Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Vers un nouvel Oracle

Performances respectives [VPP]



*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

Un second exemple

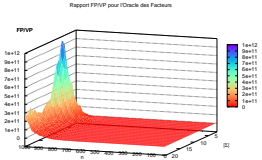
Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Vers un nouvel Oracle

Performances respectives [$\frac{EP}{VP}$]

Oracles	Facteurs	Suffixes
Version Originale		
Transitions Gardées		

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

Un second exemple

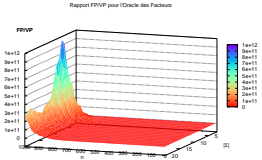
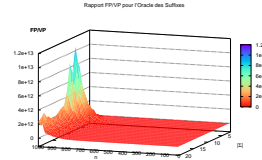
Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Vers un nouvel Oracle

Performances respectives [$\frac{FP}{VP}$]

Oracles	Facteurs	Suffixes
Version Originale		
Transitions Gardées		

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Ideé principale

Principe de la fonction de suppléance

Fonction succédanée en supplément

Nouvel algorithme « on-line »

Un second exemple

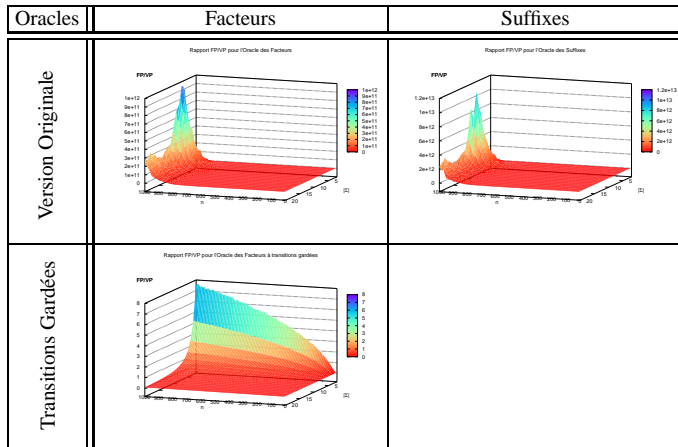
Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Vers un nouvel Oracle

Performances respectives [$\frac{EP}{VP}$]



*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Idee principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

Un second exemple

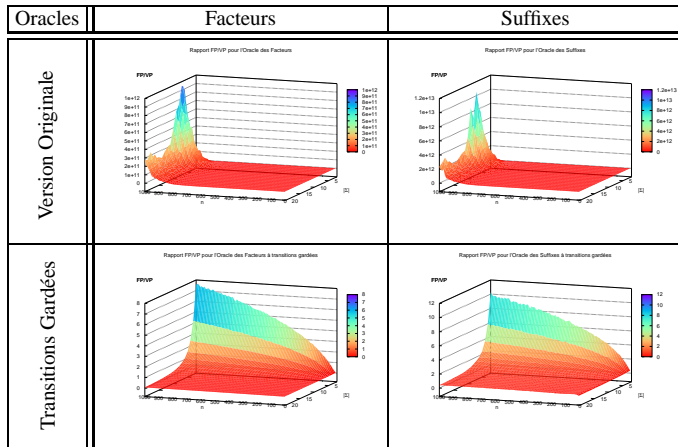
Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

Vers un nouvel Oracle

Performances respectives [$\frac{EP}{VP}$]



*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

Fiabilité des Oracles

Ideé principale

Principe de la fonction de suppléance

Fonction succédanée en suppléant

Nouvel algorithme « on-line »

Un second exemple

Critères d'évaluation

Performances respectives

*Conclusions &
perspectives*

- Oracle original :

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

- Oracle original :
 - caractérisation du langage reconnu ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

- Oracle original :
 - caractérisation du langage reconnu ;
 - mise en évidence de propriétés ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

- Oracle original :
 - caractérisation du langage reconnu ;
 - mise en évidence de propriétés ;
 -

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

- Oracle original :
 - caractérisation du langage reconnu ;
 - mise en évidence de propriétés ;
 - :
- Oracle à transitions gardées :

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

- Oracle original :
 - caractérisation du langage reconnu ;
 - mise en évidence de propriétés ;
 - :
- Oracle à transitions gardées :
 - algorithme « on-line » ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

- Oracle original :
 - caractérisation du langage reconnu ;
 - mise en évidence de propriétés ;
 - :
- Oracle à transitions gardées :
 - algorithme « on-line » ;
 - analyse expérimentale.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

- Oracles et produits dérivés :

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

- Oracles et produits dérivés :
 - proposer un automate (homogène) minimal en nombre de transitions ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

- Oracles et produits dérivés :
 - proposer un automate (homogène) minimal en nombre de transitions ;
 - proposer un automate (homogène) minimal en nombre de faux positifs ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

- Oracles et produits dérivés :
 - proposer un automate (homogène) minimal en nombre de transitions ;
 - proposer un automate (homogène) minimal en nombre de faux positifs ;
 -

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

- Oracles et produits dérivés :
 - proposer un automate (homogène) minimal en nombre de transitions ;
 - proposer un automate (homogène) minimal en nombre de faux positifs ;
 - :
- Oracle à transitions gardées :

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

- Oracles et produits dérivés :
 - proposer un automate (homogène) minimal en nombre de transitions ;
 - proposer un automate (homogène) minimal en nombre de faux positifs ;
 - :
- Oracle à transitions gardées :
 - caractérisation du langage reconnu ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

- Oracles et produits dérivés :
 - proposer un automate (homogène) minimal en nombre de transitions ;
 - proposer un automate (homogène) minimal en nombre de faux positifs ;
 - :
- Oracle à transitions gardées :
 - caractérisation du langage reconnu ;
 - analyse statistique (vs. expérimentale) ;

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

- Oracles et produits dérivés :
 - proposer un automate (homogène) minimal en nombre de transitions ;
 - proposer un automate (homogène) minimal en nombre de faux positifs ;
 -
- Oracle à transitions gardées :
 - caractérisation du langage reconnu ;
 - analyse statistique (vs. expérimentale) ;
 - intégrer l'Oracle à transitions gardées en lieu et place de l'Oracle original.

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

*Je vous remercie de votre
attention...*

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

*Je vous remercie de votre
attention...*

*...et espère que mes réponses à vos
questions vous satisferont.*

*Oracle à transitions
gardées*

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

*Langage reconnu par les
Oracles*

Vers un nouvel Oracle

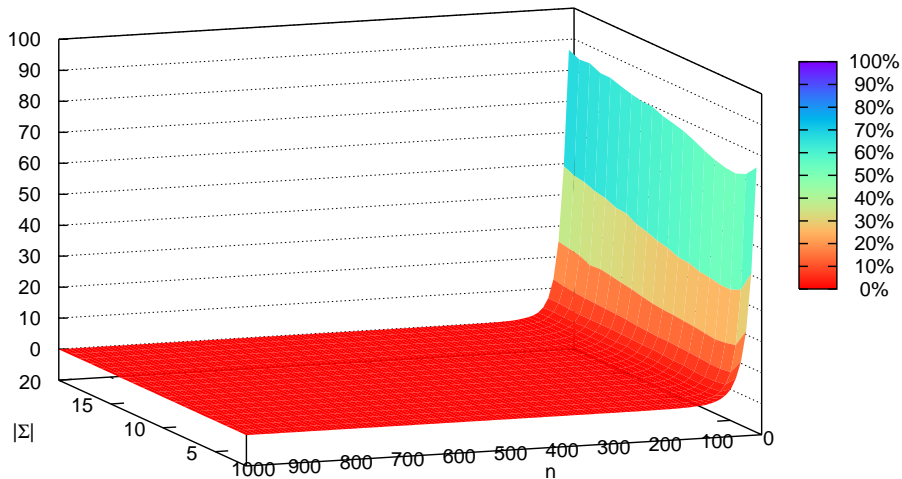
*Conclusions &
perspectives*

Ce qui a été réalisé

Ce qui reste à faire

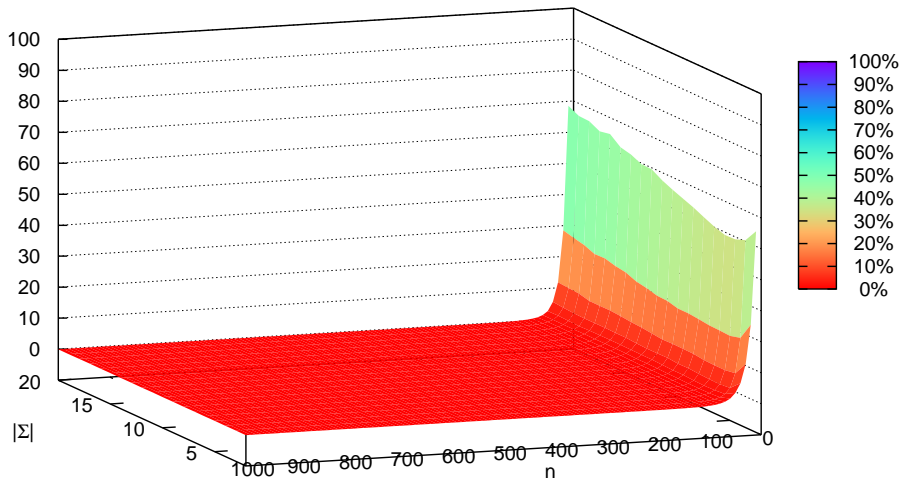
Valeur Prédicative Positive pour l'Oracle des Facteurs

VPP (en %)



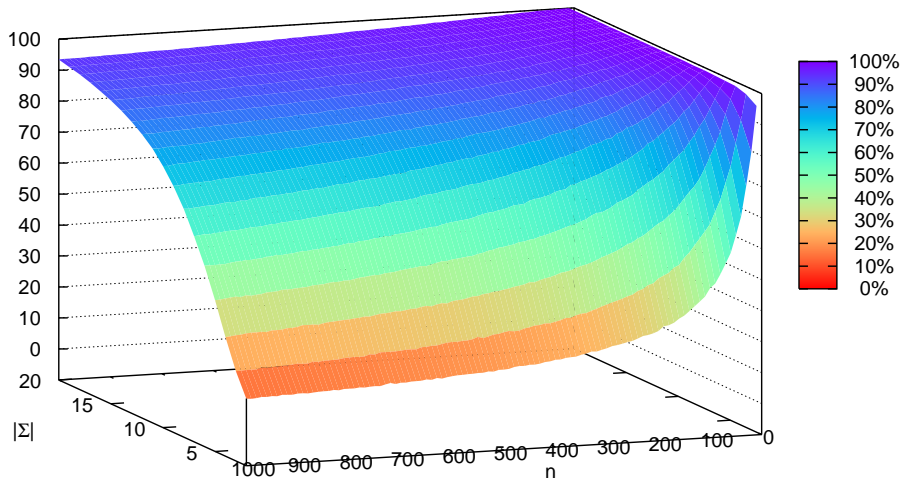
Valeur Prédicative Positive pour l'Oracle des Suffixes

VPP (en %)



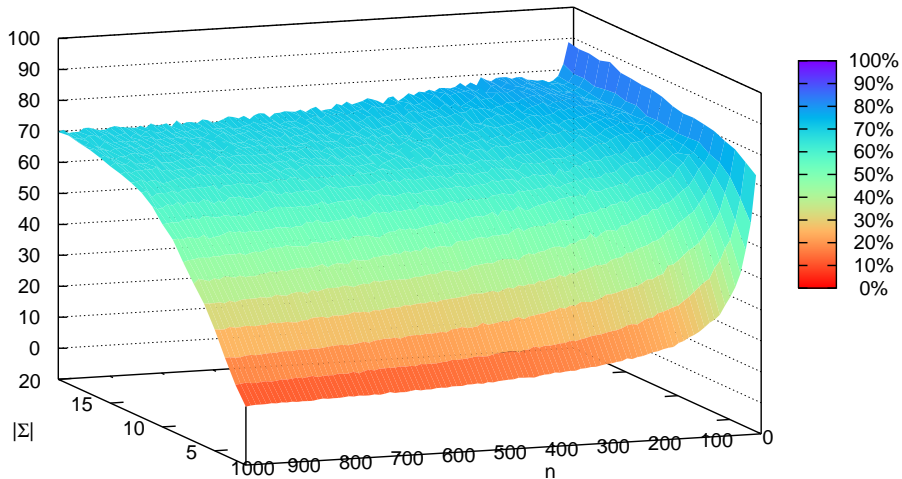
Valeur Prédicative Positive pour l'Oracle des Facteurs à transitions gardées

VPP (en %)



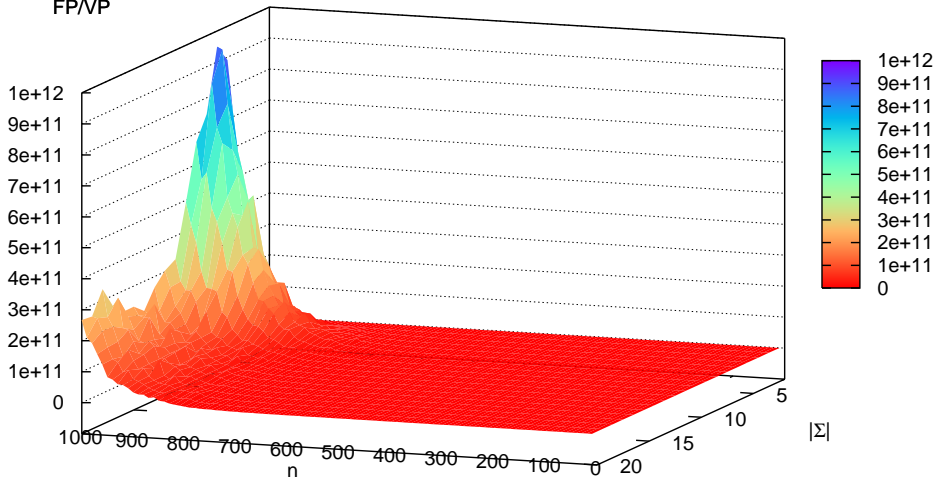
Valeur Prédicative Positive pour l'Oracle des Suffixes à transitions gardées

VPP (en %)

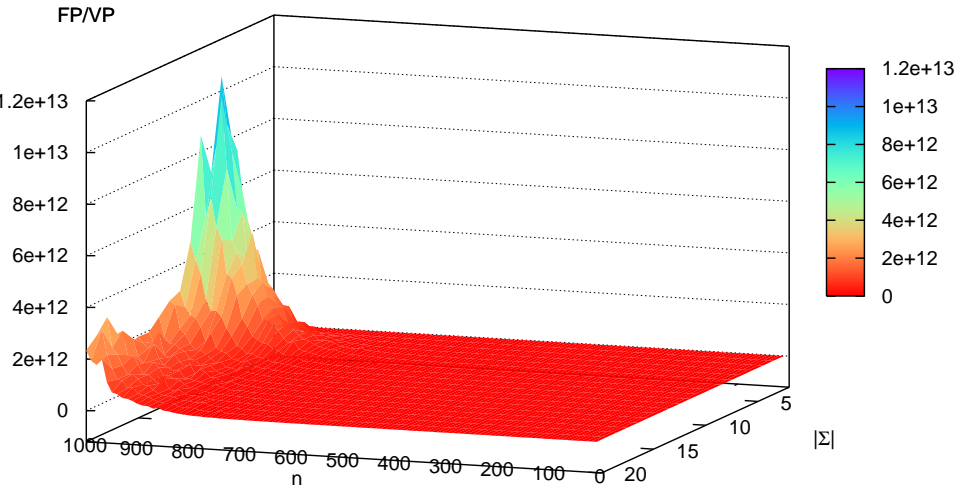


Rapport FP/VP pour l'Oracle des Facteurs

FP/VP

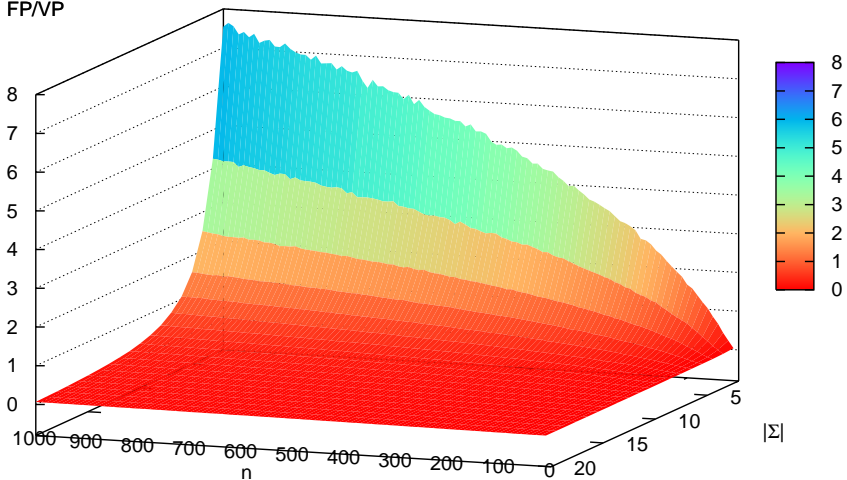


Rapport FP/VP pour l'Oracle des Suffixes



Rapport FP/VP pour l'Oracle des Facteurs à transitions gardées

FP/VP



Rapport FP/VP pour l'Oracle des Suffixes à transitions gardées

