Département d'informatique et de génie logiciel

Compression de données
IFT-4003/IFT-7023

Notes de cours
Compression basée sur les contextes

Édition Hiver 2012

Mohamed Haj Taieb

Local: PLT 2113

Courriel: mohamed.haj-taieb.1@ulaval.ca

Faculté des sciences et de génie
Département de génie électrique et de génie informatique

UNIVERSITÉ LAVAL

Plan de la présentation

- Compression basée sur les contextes:
  - Introduction
  - Prédiction avec correspondance partielle
  - Symbole échappe
  - Longueur du contexte
  - Principe d'exclusion
  - La transformation Burrows-Wheeler
  - Encodeur associatif de Buyanovsky
  - Compression dynamique de Markov

Introduction

- Compression basée sur les contextes
  - Utilisation d'un minimum d'hypothèses sur les statistiques des données.
  - Utilisation du contexte des données actuelles et passées.
  - → Compression de texte.
- Constatation
  - Probabilités débalancées → plus de compression.
  - Cas extrême: P(1)=1 et P(0)=0 → H=0.
  - L'idée est alors de trouver un moyen pour représenter les données avec plus de débalancement.
  - → observation de la probabilité du symbole dans le contexte courant: observation de l'historique.

Entropie de la langue anglaise selon Shannon

- Alphabet de 26 lettres:
  - $H_0 = \log_2(26) \approx 4.7$ bits/lettre
  - En considérant la distribution de chaque lettres:
    - $H_1 \approx 4.14$ bits/lettre
  - On se base sur l'observation de la lettre précédente
    - $H_2 \approx 3.56$ bits/lettre [basé sur les digrammes]
  - On se base sur l'observation des 2lettres précédentes
    - $H_3 \approx 3.3$ bits/lettre [basé sur les trigrammes]
- Entropie d'un mot:
  - $H_{word} = -\sum_{n=1..8727} p_n \log_2 p_n = 11.82$ bits/mots
  - Longueur moyenne d'un mot = 4.5
  - $H_{word} = 11.82/4.5 = 2.62$ bits/lettre

Formule de zif sur la fréquence de mot: $p_n=0.1/n$

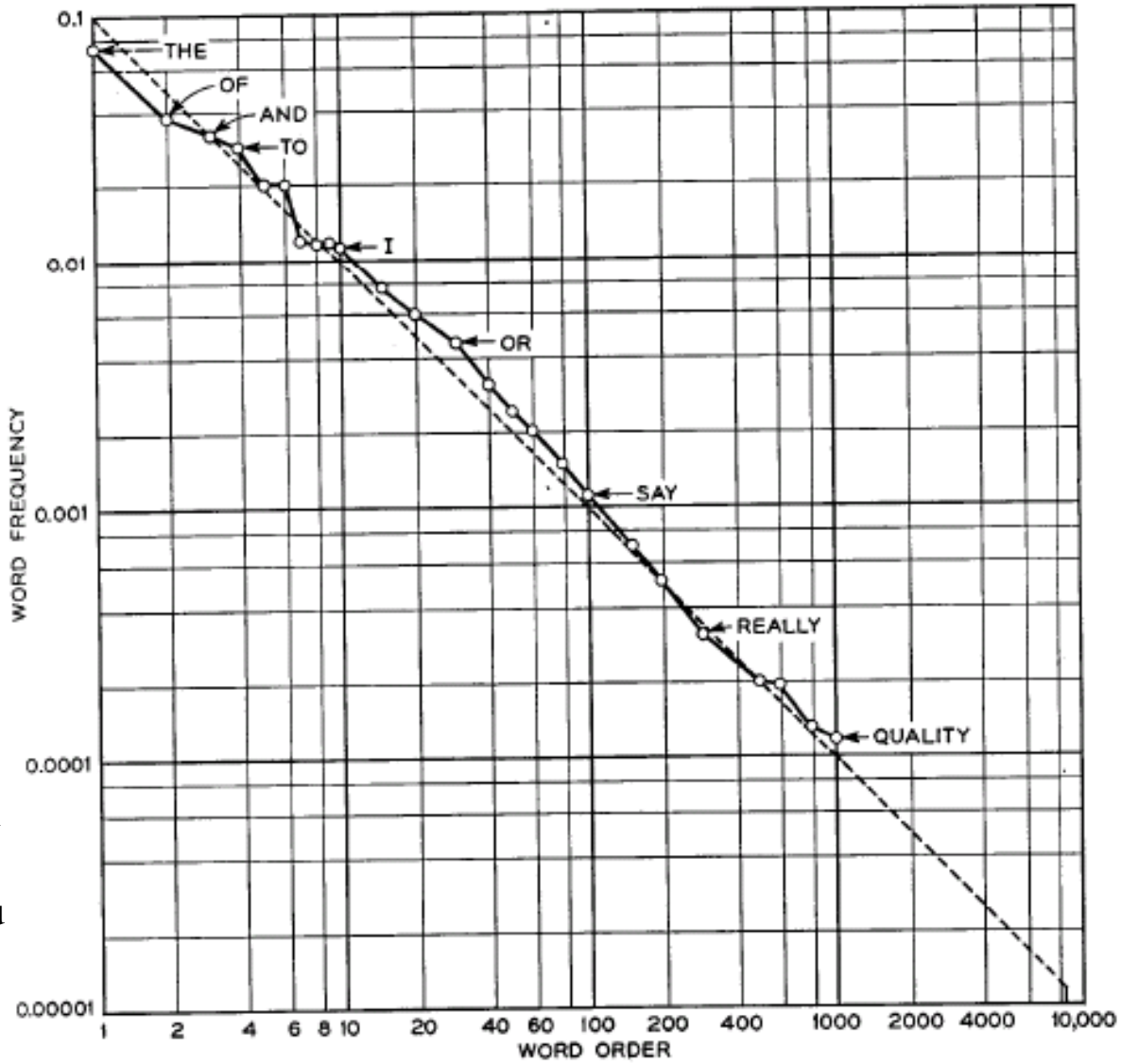


Fig. 1—Relative frequency against rank for English words.

Prediction and Entropy of printed english [Claude E. Shannon] http://csc.ucdavis.edu/~cmg/Group/readings/CES-PredictEntEng.pdf

Compression basée sur la prédiction

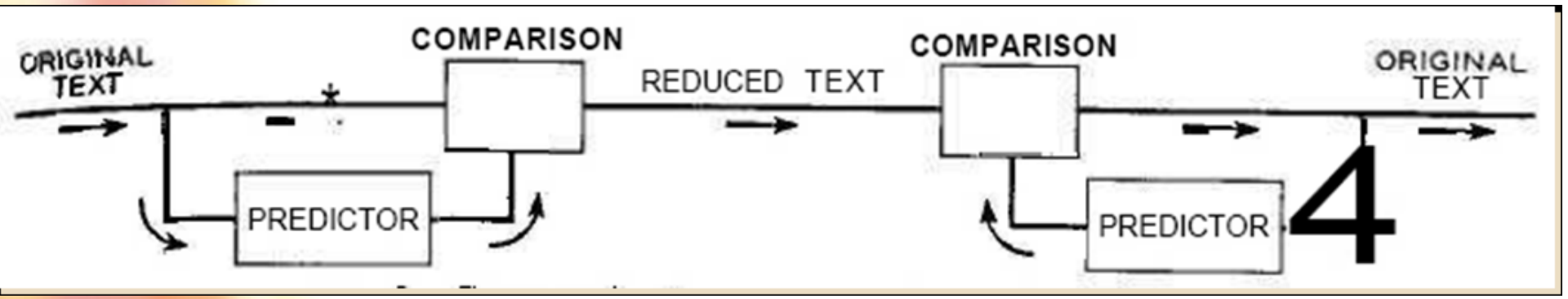
- Expérience effectuée par Shannon: réussite =70%

```
Text:  THE ROOM WAS NOT VERY LIGHT A SMALL OBLONG
Human: ----ROOM-----NOT-V-----I------S-----OBL---

Text:  READING LAMP ON THE DESK SHED GLOW ON
Human: REA----------O------D----SHED-GLO--O-

Text:  POLISHED WOOD BUT LESS ON THE SHABBY RED CARPET
Human: P-L-S-----O---BU--L-S--O------SH-----RE--C-----
```

- Idée de compression de Shannon



Codage prédictif

- Pas d'intervension humaine:
  - On utilise la probabilité conditionnelle
  - Exemple: P('h')=5%, P('u')=2%
  - Mais P('h'|'t')=30% et P('u'|'q')=99%
- Contexte d'ordre supérieure:
  - Contexte d'ordre 1: P('a'), P('b'), … P('z') : 26 contextes
  - Contexte d'ordre 3: P('e'|'t','h')>P('e'|'h')> P('2'): 26^3
  - Contexte d'ordre 5: Pr(a|'p', 'r', 'o', 'b') : $26^5 = 11.8x10^7$
- Problème:
  - Quantité colossale d'information à stocker.
  - Solution: un ensemble d'algorithme: ppm (prediction with partial match).

Prédiction avec correspondance partielle

- Principe général du ppm
  - Estimation des probabilités conditionnelles au fur et à mesure de l'encodage.
  - Stockage des contextes vus dans le passé uniquement.
  - Encodage de lettre nouvelles au début, surtout au début.
  - Utilisation du symbole échappe pour signaler une nouvelle entrée.
- Remarques:
  - Pas besoin de stocker tous les contextes possibles.
  - Plusieurs méthodes ont été proposées.

Algorithme [Cleary and Witten 1984]

- Taille alphabet=M
  1. Taille du contexte = N [typiquement égale 5]
  2. Tant que nouveau symbole pour ce contexte et N>-1
     - → envoi du symbole Échappe.
     - → N–
     - → Boucler à 2
  3. Si N=-1 [Contexte d'ordre -1 contient tout les symboles]
     - → Encodage du nouveau symbole Pr=1/M
  4. Si N>-1
     - → Mise à jour des tables des compteurs
     - → Encodage arithmétique

Exemple PPM méthode A: ppma [1]

- Séquence: **this-is**-the-tithe
  - Contexte maximale N=2.
  - Codage arithmétique m=6: u=111111=63 et l=000000=0
  - État initial: encodage des 7 premiers caractères.

Contexte d'ordre -1		
Lettre	**Count**	**Cum_count**
t	1	1
h	1	2
i	1	3
s	1	4
e	1	5
-	1	6
Total count		6

Contexte d'ordre zéro		
Lettre	**Count**	**Cum_count**
t	1	1
h	1	2
i	2	4
s	2	6
-	1	7
<Esc>	1	8
Total count		8

Exemple PPM méthode A: ppma [2]

- Séquence: **this-is**-the-tithe

Contexte d'ordre 1				
Contexte	Lettre	Count	CC	TC
t	h	1	1	
	Esc	1	2	2
h	i	1	1	
	Esc	1	2	2
i	s	2	2	
	Esc	1	3	3
s	-	1	1	
	Esc	1	2	2
-	s	1	1	
	Esc	1	2	2

Contexte d'ordre 2				
Contexte	Lettre	Count	CC	TC
th	i	1	1	
	Esc	1	2	2
hi	s	1	1	
	ESC	1	2	2
is	-	1	1	
	Esc	1	2	2
s-	i	1	1	
	ESC	1	2	2
-i	s	1	1	
	Esc	1	2	2

Exemple PPM méthode A: ppma [3]

- Séquence: this-**is-**the-tithe
  - 'is -' ∈ contexte d'ordre 2: mise à jour de cette table
  - l=000000 et u=111111

Contexte	Lettre	Count	CC	TC
is	**-**	**1**	**1**	
	Esc	1	2	2

$$u = 0 + \left\lfloor (63 - 0 + 1) \times \frac{1}{2} \right\rfloor - 1 = 31 = (011111)_2$$

$$l = 0 + \left\lfloor (63 - 0 + 1) \times \frac{0}{2} \right\rfloor = 0 = (000000)_2$$

  - l=**0**00000, u=**0**11111 → env. MSB
  - E1→ l=000000, u= 111111
  - Envoi: **0**

Contexte d'ordre 2

Contexte	Lettre	Count	CC	TC
th	i	1	1	
	Esc	1	2	2
hi	s	1	1	
	ESC	1	2	2
is	**-**	1→2	1→2	
	Esc	1	2→3	2→3
s-	i	1	1	
	ESC	1	2	2
-i	s	1	1	
	Esc	1	2	2

Exemple PPM méthode A: ppma [4]

- Séquence: this-**is**-the-tithe
  - Mise à jour des tables des contextes d'ordre 1 et d'ordre 0

Contexte d'ordre zéro		
Lettre	**Count**	**Cum_count**
t	1	1
h	1	2
i	2	4
s	2	6
-	**1→2**	**7→8**
<Esc>	1	**8→9**
Total count		**8→9**

Contexte d'ordre 1				
Contexte	Lettre	Count	CC	TC
t	h	1	1	
	Esc	1	2	2
h	i	1	1	
	Esc	1	2	2
i	s	2	2	
	Esc	1	3	3
s	-	1→2	1→2	
	Esc	1	2→3	2→3
-	s	1	1	
	Esc	1	2	2

- Envoi: **0**

Exemple PPM méthode A: ppma [5]

- Séquence: this-i**s-t**he-tithe
- Encodage arithmétique de 't': l=000000=0 et u=111111=63

'**s-** t' n'existe pas→N- (voir le contexte d'ordre 1) et envoi de <esc>

$$u = 0 + \left\lfloor (63 - 0 + 1) \times \frac{2}{2} \right\rfloor - 1 = 63 = (111111)_2$$

$$l = 0 + \left\lfloor (63 - 0 + 1) \times \frac{1}{2} \right\rfloor = 32 = (100000)_2$$

- l=**1**00000, u=**1**11111 → env. MSB
- E2→ l=000000, u= 111111
- Il faut aussi faire la mise à jour de cette table.
- Envoi: 0**1**

Contexte d'ordre 2				
Contexte	Lettre	Count	CC	TC
th	i	1	1	
	Esc	1	2	2
hi	s	1	1	
	ESC	1	2	2
is	-	2	2	
	Esc	1	3	3
s-	i	1	1	
	ESC	**1**	**2**	**2**
-i	s	1	1	
	Esc	1	2	2

Exemple PPM méthode A: ppma [6]

- Séquence: this-i**s-t**he-tithe
- Mise à jour de la table de contexte d'ordre 2
- On passe au contexte d'ordre 1

Contexte d'ordre 2				
Contexte	Lettre	Count	CC	TC
th	i	1	1	
	Esc	1	2	2
hi	s	1	1	
	ESC	1	2	2
is	-	2	2	
	Esc	1	3	3
s-	i	1	1	
	t	1	2	
	ESC	**1**	**3**	**3**
-i	s	1	1	
	Esc	1	2	2

- Envoi: 0**1**

Exemple PPM méthode A: ppma [7]

- Séquence: this-is-**t**he-tithe
  - Encodage arithmétique de 't': l=000000=0 et u=111111=63

'- t' n'existe pas→N- (voir le contexte d'ordre 0) et envoi de <esc>

$$u = 0 + \left\lfloor (63 - 0 + 1) \times \frac{2}{2} \right\rfloor - 1 = 63 = (111111)_2$$

$$l = 0 + \left\lfloor (63 - 0 + 1) \times \frac{1}{2} \right\rfloor = 32 = (100000)_2$$

- l=**1**00000, u=**1**11111 → env. MSB
- E2→ l=000000, u= 111111
- Il faut aussi faire la mise à jour de cette table.
- Envoi: 01**1**

Contexte d'ordre 1				
Contexte	Lettre	Count	CC	TC
t	h	1	1	
	Esc	1	2	2
h	i	1	1	
	Esc	1	2	2
i	s	2	2	
	Esc	1	3	3
s	-	2	2	
	Esc	1	3	3
-	**s**	**1**	**1**	
	Esc	**1**	**2**	**2**

Exemple PPM méthode A: ppma [8]

- Séquence: this-i**s-t**he-tithe
- Mise à jour de la table de contexte d'ordre 1
- On passe au contexte d'ordre 0

Contexte d'ordre 1				
Contexte	Lettre	Count	CC	TC
t	h	1	1	
	Esc	1	2	2
h	i	1	1	
	Esc	1	2	2
i	s	2	2	
	Esc	1	3	3
s	-	2	2	
	Esc	1	3	3
-	**s**	**1**	**1**	
	t	**1**	**2**	
	Esc	**1**	**3**	**3**

- Envoi: 011

Exemple PPM méthode A: ppma [9]

- Séquence: this-is-**t**he-tithe
- Encodage arithmétique de 't' l=000000=0 et u=111111=63
- Mise à jour tables ordre 0.

t existe → Count t++

$$u = 0 + \left\lfloor (63 - 0 + 1) \times \frac{1}{9} \right\rfloor - 1 = 6 = (000110)_2$$

$$l = 0 + \left\lfloor (63 - 0 + 1) \times \frac{0}{9} \right\rfloor = 0 = (000000)_2$$

Contexte d'ordre zéro		
Lettre	**Count**	**Cum_count**
t	1→2	1→2
h	1	2→3
i	2	4→5
s	2	6→7
-	2	8→9
<Esc>	1	9→10
Total count		9→10

- l=**000**000, u=**000**110 → env. 000
- 3xE1→ l=000000, u= 110111
- Envoi: 011**000**

Exemple PPM méthode A: ppma [10]

- Séquence: this-is-the-tithe

Le contexte du second ordre '**-t**' n'existe pas→on passe directement au premier ordre et on effectue la mise à jour.

Contexte d'ordre 2				
Contexte	Lettre	Count	CC	TC
th	i	1	1	
	Esc	1	2	2
hi	s	1	1	
	ESC	1	2	2
is	-	2	2	
	Esc	1	3	3
s-	i	1	1	
	t	1	2	
	ESC	1	3	3
-i	s	1	1	
	Esc	1	2	2

- Envoi: 011000

Exemple PPM méthode A: ppma [11]

- Séquence: this-is-the-tithe

Ajout du contexte '**-t h**'

Contexte d'ordre 2				
Contexte	Lettre	Count	CC	TC
th	i	1	1	
	Esc	1	2	2
hi	s	1	1	
	ESC	1	2	2
is	-	2	2	
	Esc	1	3	3
s-	i	1	1	
	t	1	2	
	ESC	1	3	3
-i	s	1	1	
	Esc	1	2	2
-t	h	1	1	
	Esc	1	2	2

- Envoi: 011000

Exemple PPM méthode A: ppma [12]

- Séquence: this-is-the-tithe
  - Encodage arithmétique de 't' l=000000=0 et u=110111=55

$$u = 0 + \left\lfloor (55 - 0 + 1) \times \frac{1}{2} \right\rfloor - 1 = 27 = (011011)_2$$

$$l = 0 + \left\lfloor (55 - 0 + 1) \times \frac{0}{2} \right\rfloor = 0 = (000000)_2$$

- l=**0**000000, u=**0**11011 → env. 0
- E1→ l=000000, u= 110111
- Envoi: 011000**0**

t h existe → count ++

Contexte d'ordre 1				
Contexte	Lettre	Count	CC	TC
t	**h**	**1→2**	**1→2**	
	Esc	**1**	**2→3**	**2→3**
h	i	1	1	
	Esc	1	2	2
i	s	2	2	
	Esc	1	3	3
s	-	1	1	
	Esc	1	2	2
-	s	1	1	
	Esc	1	2	2
-	t	1	1	
	Esc	1	2	2

Exemple PPM méthode A: ppma [13]

- Séquence: this-is-the-tithe
  - Mise à jour de la table du contexte d'ordre zéro

Et ainsi de suite …

Contexte d'ordre zéro		
Lettre	**Count**	**Cum_count**
t	2	2
h	1→2	3→4
i	2	5→6
s	2	7→8
-	2	9→10
<Esc>	1	10→11
Total count		10→11

- Envoi: 0110000

Le symbole <Esc>

- ppma
  - On rajoute un symbole <Esc> pour chaque contexte.
  - Total count (TC) est dilaté de 1.
- Ppmb [variante de ppma]
  - Nouvelle entrée à la table → count de <ESC> ++
  - Un symbole nouveau → initialisation à 0 au lieu de 1.
  - TC est égale au nombre occurrence totale dans le contexte.

Contexte d'ordre 4 ppma			
Contexte	Lettre	Count	CC
prob	a	10	10
	l	9	19
	o	3	22
	ESC	1	23
Total count			23

Contexte d'ordre 4 ppmb			
Contexte	Lettre	Count	CC
prob	a	9	9
	l	8	17
	o	2	19
	ESC	3	22
Total count			22

Exemple: ppmb

- État initial: 9 x probability+ 8 x problem
- Encodage: probability probosics problem probosics probosics

État initial

Contexte	Lettre	Count
prob	a	8
	l	7
	ESC	2
Total count		17

probability

Contexte	Lettre	Count
prob	**a**	**9**
	l	7
	ESC	2
Total count		18

probosics

Contexte	Lettre	Count
prob	a	9
	l	7
	o	**0**
	ESC	**3**
Total count		19

probolem

Contexte	Lettre	Count
prob	a	9
	l	**8**
	o	0
	ESC	3
Total count		20

probosics

Contexte	Lettre	Count
prob	a	9
	l	8
	o	**1**
	ESC	3
Total count		21

probosics

Contexte	Lettre	Count
prob	a	9
	l	8
	o	**2**
	ESC	3
Total count		22

ppmc

- Ppmb [variante de ppma]
  - Nouvelle entrée à la table → count de <ESC> ++
  - Un symbole nouveau → initialisation à 0 au lieu de 1.
  - TC est égale au nombre occurrence totale dans le contexte.
- Ppmc [variante de ppmb]
  - Nouvelle entrée à la table → count de <ESC> ++
  - Un symbole nouveau → initialisation à 1.
  - TC est dilaté chaque fois il y a une nouvelle entrée.

Exemple: ppmc

- État initial: 9 x probability+ 8 x problem
- Encodage: probability probosics problem probosics probosics

État initial

Contexte	Lettre	Count
prob	**a**	**9**
	l	**8**
	ESC	2
Total count		**19**

probability

Contexte	Lettre	Count
prob	**a**	**10**
	l	8
	ESC	2
Total count		20

probosics

Contexte	Lettre	Count
prob	a	10
	l	8
	o	**1**
	ESC	**3**
Total count		22

probolem

Contexte	Lettre	Count
prob	a	10
	l	**9**
	o	1
	ESC	3
Total count		23

probosics

Contexte	Lettre	Count
prob	a	10
	l	9
	o	**2**
	ESC	3
Total count		24

probosics

Contexte	Lettre	Count
prob	a	10
	l	9
	o	**3**
	ESC	3
Total count		25

Principe d'exclusion

- Codage arithmétique
  - Sous-intervalle réduit → application E1, E2 et E3 → plus de bits sont nécessaires.
  - Si l'on réduit le nombre de symboles → sous-intervalles plus large → moins de bits sont nécessaire
- Le principe d'exclusion
  - Détecter les situations où l'on peut éliminer certains symboles des tables de contexte.
  - Lors du passage d'un contexte d'ordre k à un autre contexte d'ordre k-1 les symboles communs peuvent être éliminés.

Exemple: Principe d'exclusion

État initial
Contexte d'ordre 2

Contexte	Lettre	Count
ob	**l**	10
	o	3
	ESC	2
Total count		15

État initial
Contexte d'ordre 1

Contexte	Lettre	Count
b	**l**	5
	o	3
	a	4
	r	2
	e	2
	ESC	5
Total count		21

Encodage: de la lettre 'a':

On examine au départ le contexte d'ordre 2.

La lettre 'a' n y figure pas → envoi de <Esc>.

<Esc> → on sait alors que ce n'est ni 'l' ni 'o'.

Lors du passage au contexte d'ordre 1 on peut éliminer les entrées 'l' et 'o'.

État initial
Contexte d'ordre 1

Contexte	Lettre	Count
b	a	4
	r	2
	e	2
	ESC	**3**
Total count		**11**

Codage arithmétique en utilisant les counts du nouveau tableau.

Utilisation de l'ancien tableau pour la mise à jour.

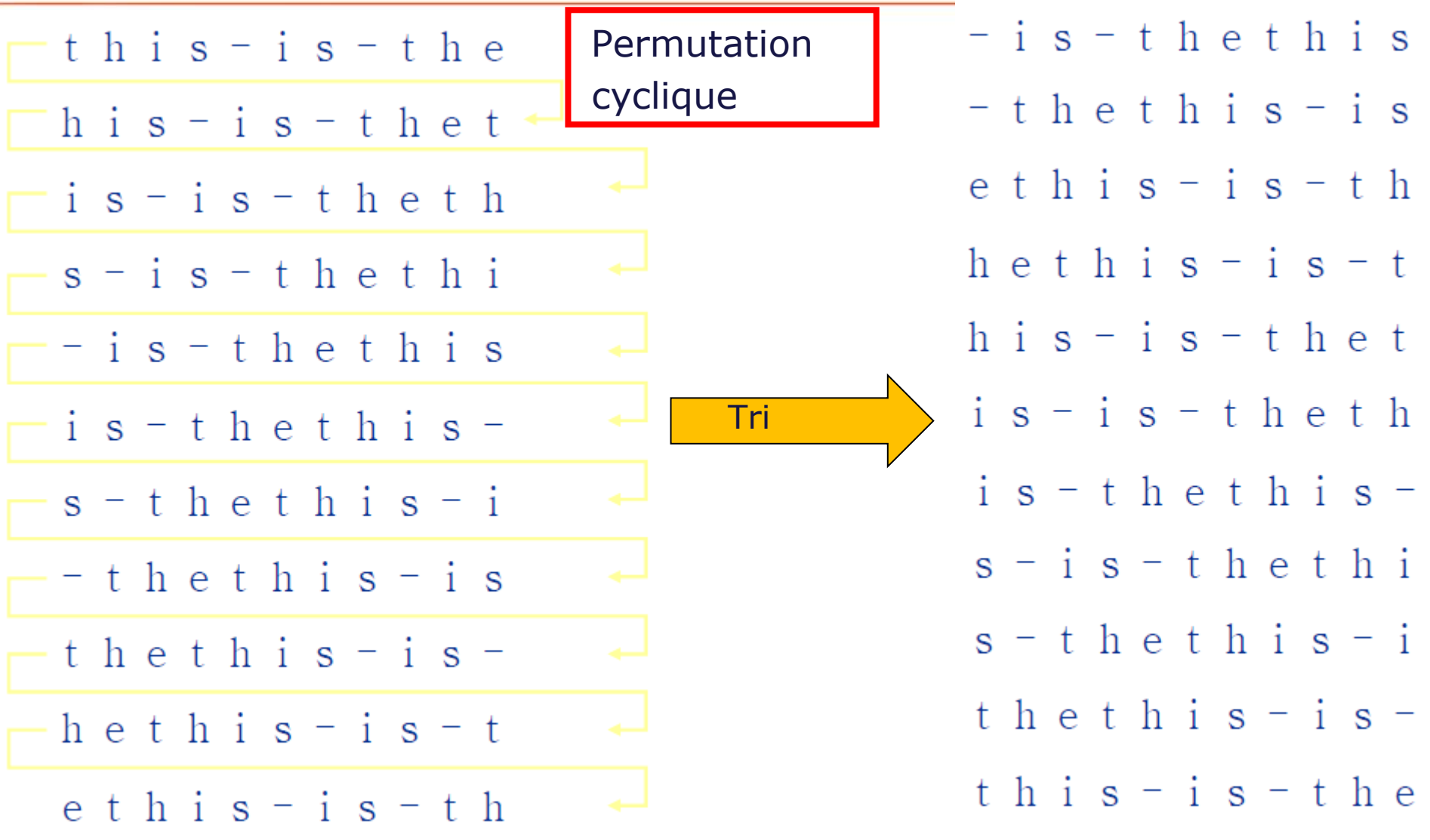
La transformation de Burrows-Wheeler (1)

- Algorithme de BWT
  - Utilisation du contexte du symbole encodé.
  - Il faut que toute la séquence soit disponible dés le début.
  - Transformation d'une séquence de N symboles en une autre séquence de N symboles.
  - La raison en est que la nouvelle séquence est probablement plus appropriée pour d'autres méthodes de compression.
- Méthodes de compression
  - MTF
  - RLE
  - Huffman

La transformation de Burrows-Wheeler (2)

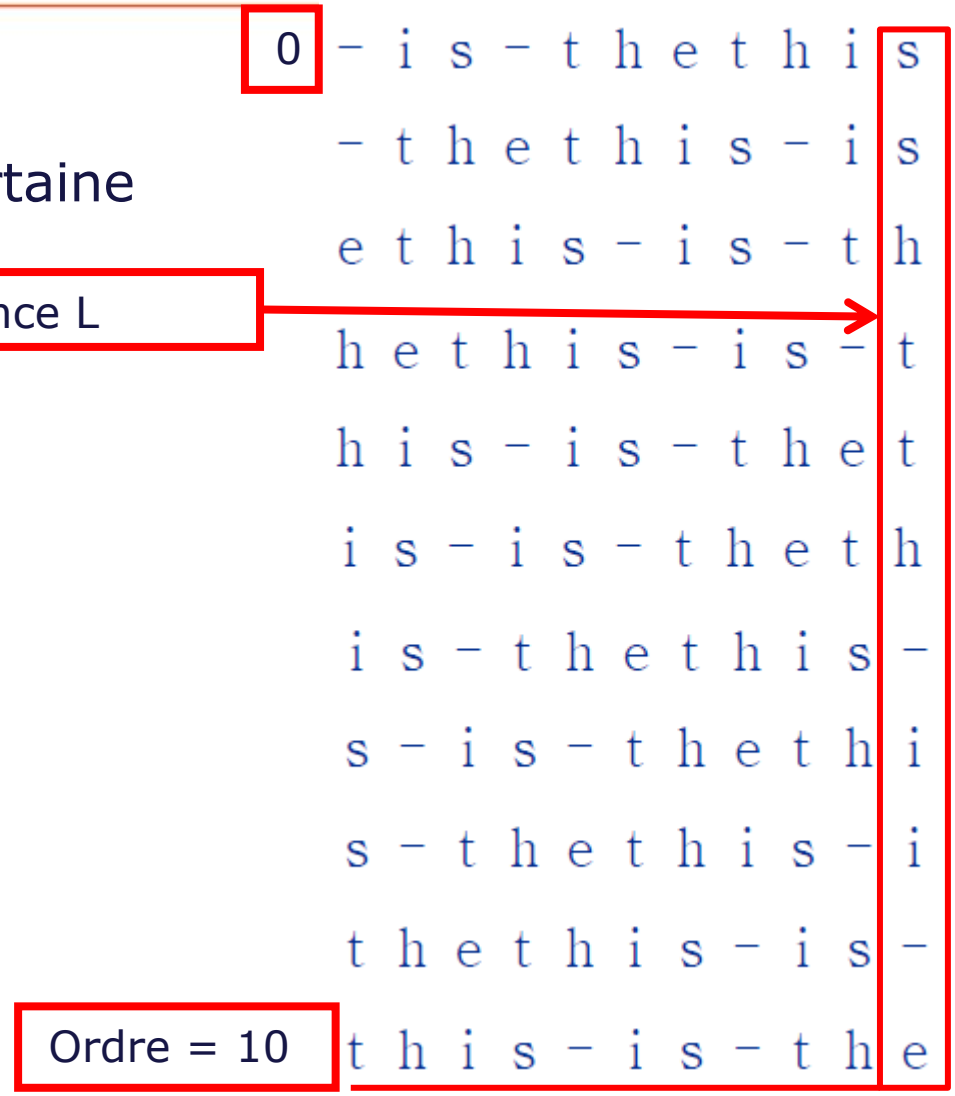
- Transformation de la séquence de taille N
  - Création de N-1 séquences à partir d'un décalage cyclique.
  - Arrangement de ces N-1 séquences avec la séquence originale selon un ordre lexicographique.
  - Sélection des dernières lettres de chaque séquence → L
  - Transmission de cette séquence L (de taille N).
  - Transmission l'ordre de la séquence originale dans l'arrangement.
  - À partir de la séquence L et de l'ordre on peut reconstituer la séquence originale.
  - Séquence taille N: N+1 éléments transmis.
  - La nouvelle séquence présente une structure particulière.

Exemple: La transformation BWT



Exemple: Encodage BWT

- Séquence L
  - On voit bien un certaine structure
- Encodage
  - 'sshtth-ii-e' , 10



Décodage BWT (1)

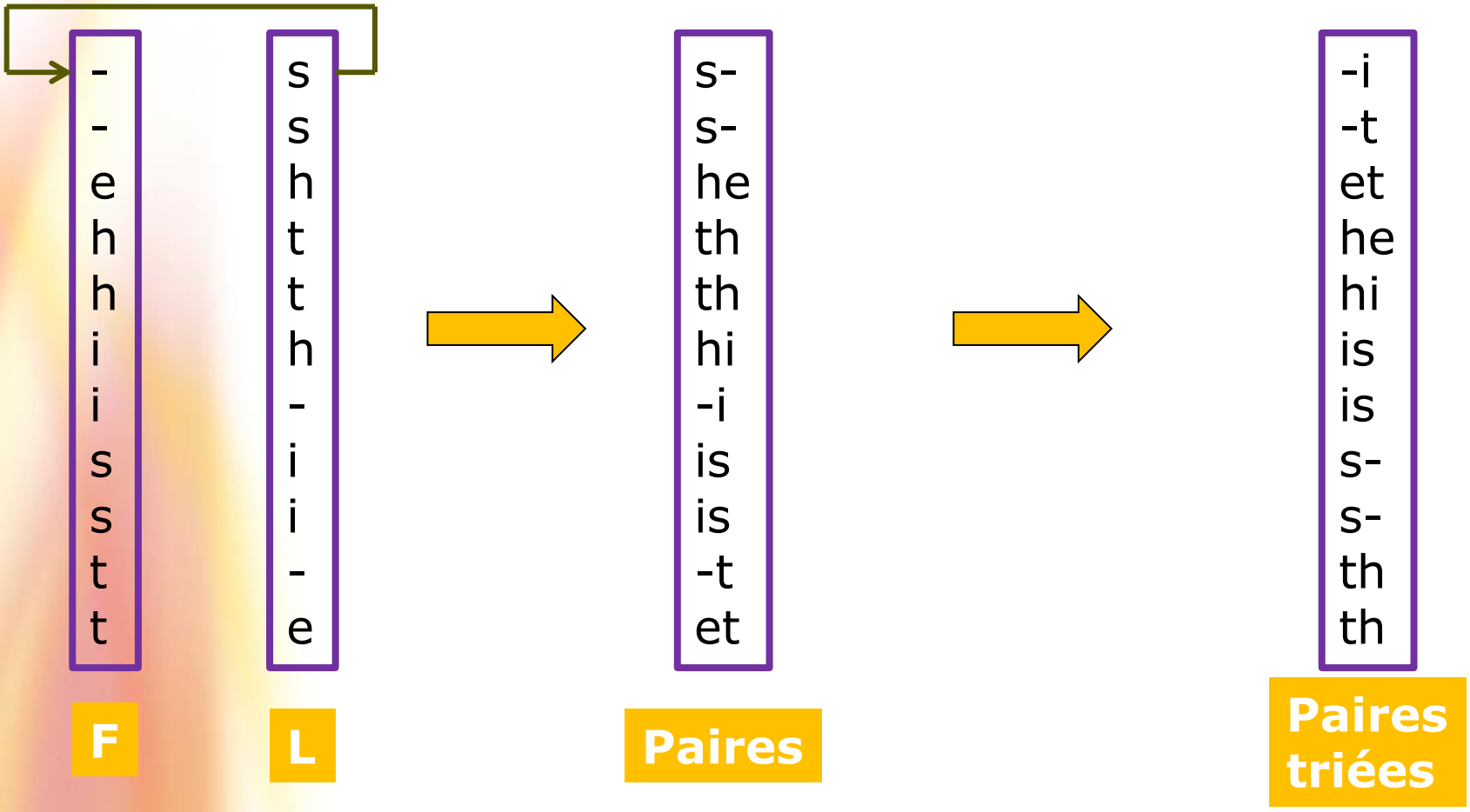
- Encodage
  - 'sshtth-ii-e' , 10
  - On voit qu'on la transformation résulte en une séquence de même taille que l'originale + un indice→ expansion.
  - Sauf que il y a des répétitions dans la séquence.
  - Ce phénomène est plus frappant pour une séquence plus longue.
- Séquence reçue: L= sshtth-ii-e
  - Génération de la séquence F qui comporte les premiers éléments de chaque ligne après le tri.
  - F est simplement le résultat du tri de la séquence L.
  - F=--ehiisstt

Décodage BWT (2)

- Observation
  - Génération de F(First) → il suffit de trier L
  - On constate que la première colonne F et la dernière colonne L(Last) forment toutes les paires possibles.
- Algorithme: approche ajout-tri
  - Répéter:
    - Ajouter une copie de L(Last)
    - Trier
  - Jusqu'à obtenir une longueur = N
  - L'indice nous indique quelle ligne choisir.

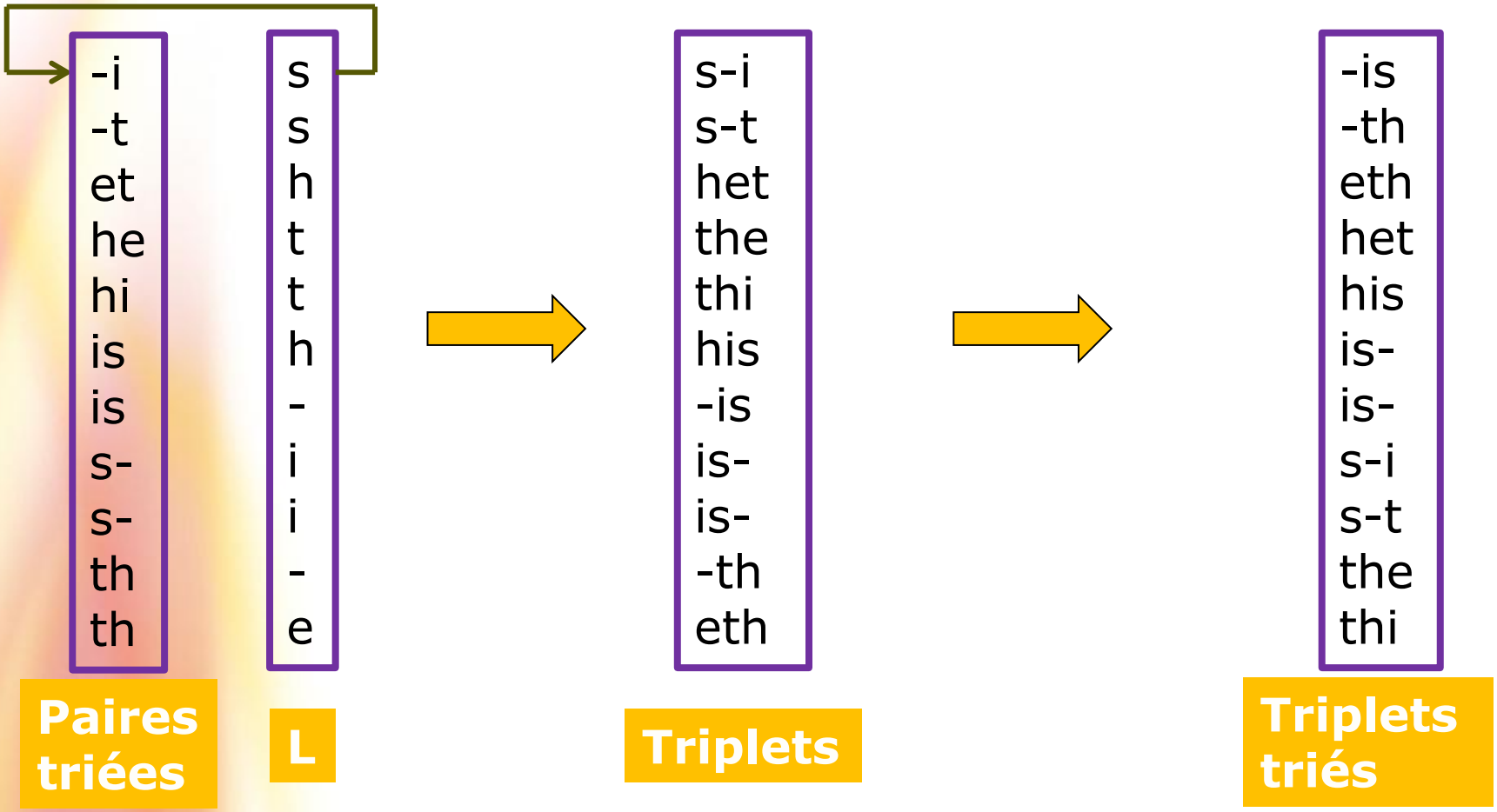
Exemple: Décodage BWT (1)

- Séquence: L= sshtth-ii-e, Indice: 10



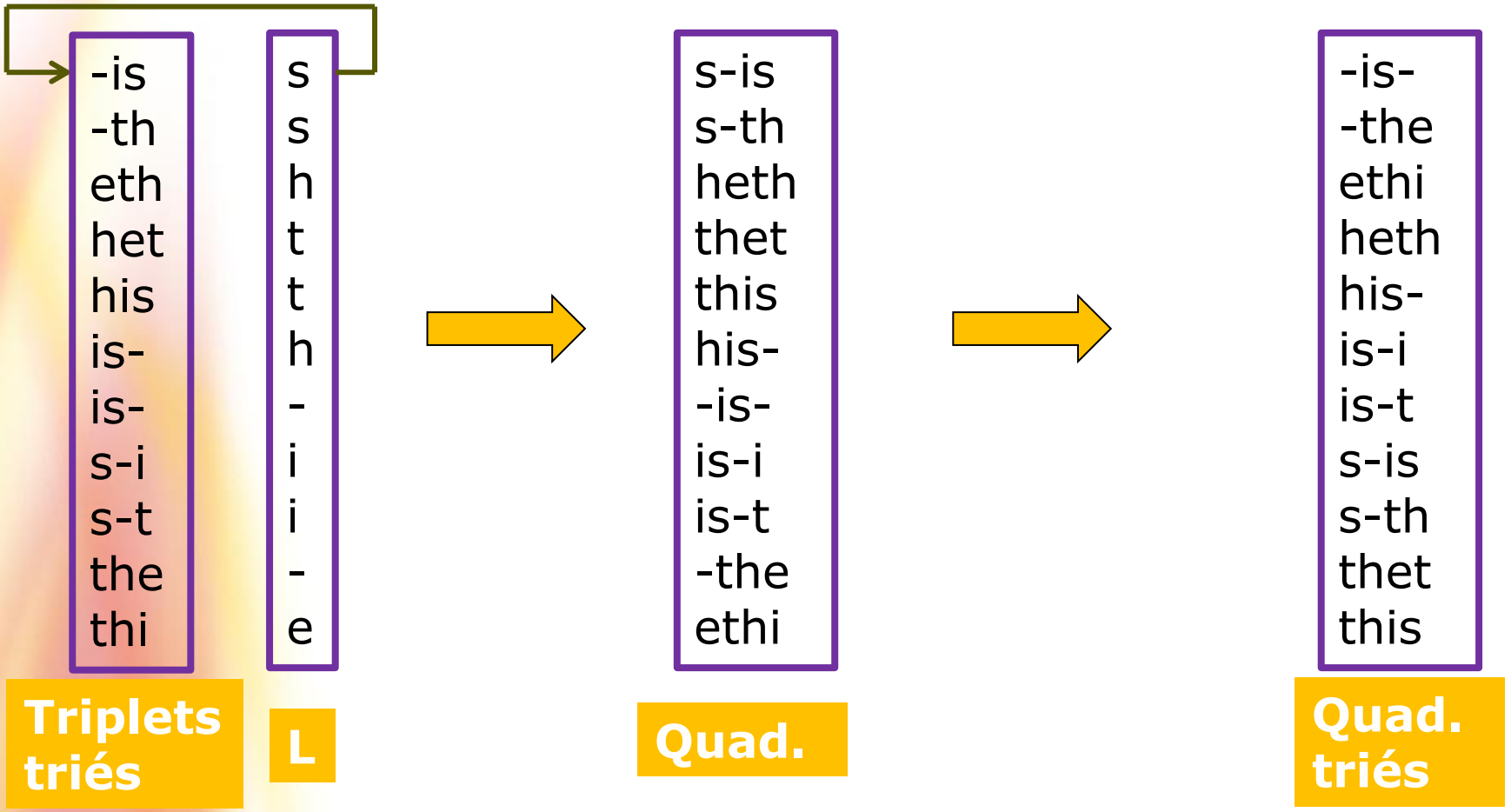
Exemple: Décodage BWT (2)

- Séquence: L= sshtth-ii-e, Indice: 10



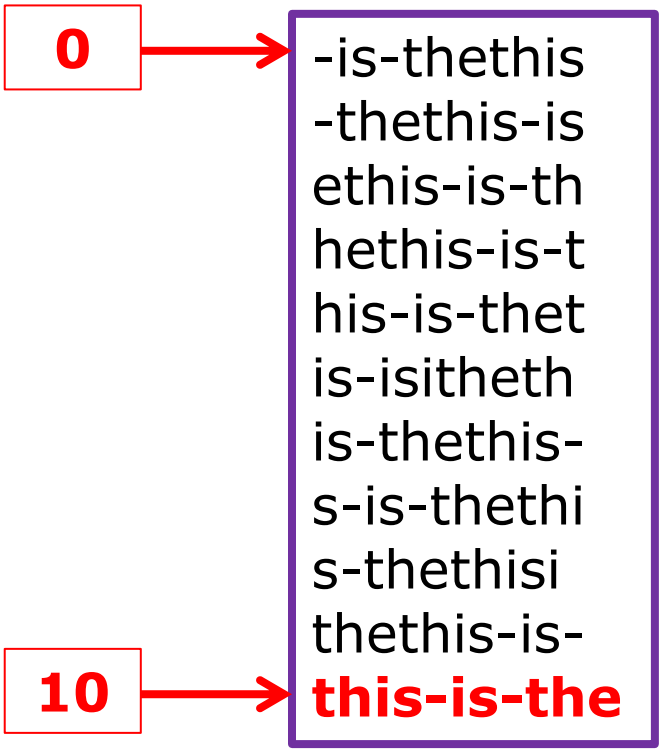
Exemple: Décodage BWT (3)

- Séquence: L= sshtth-ii-e, Indice: 10



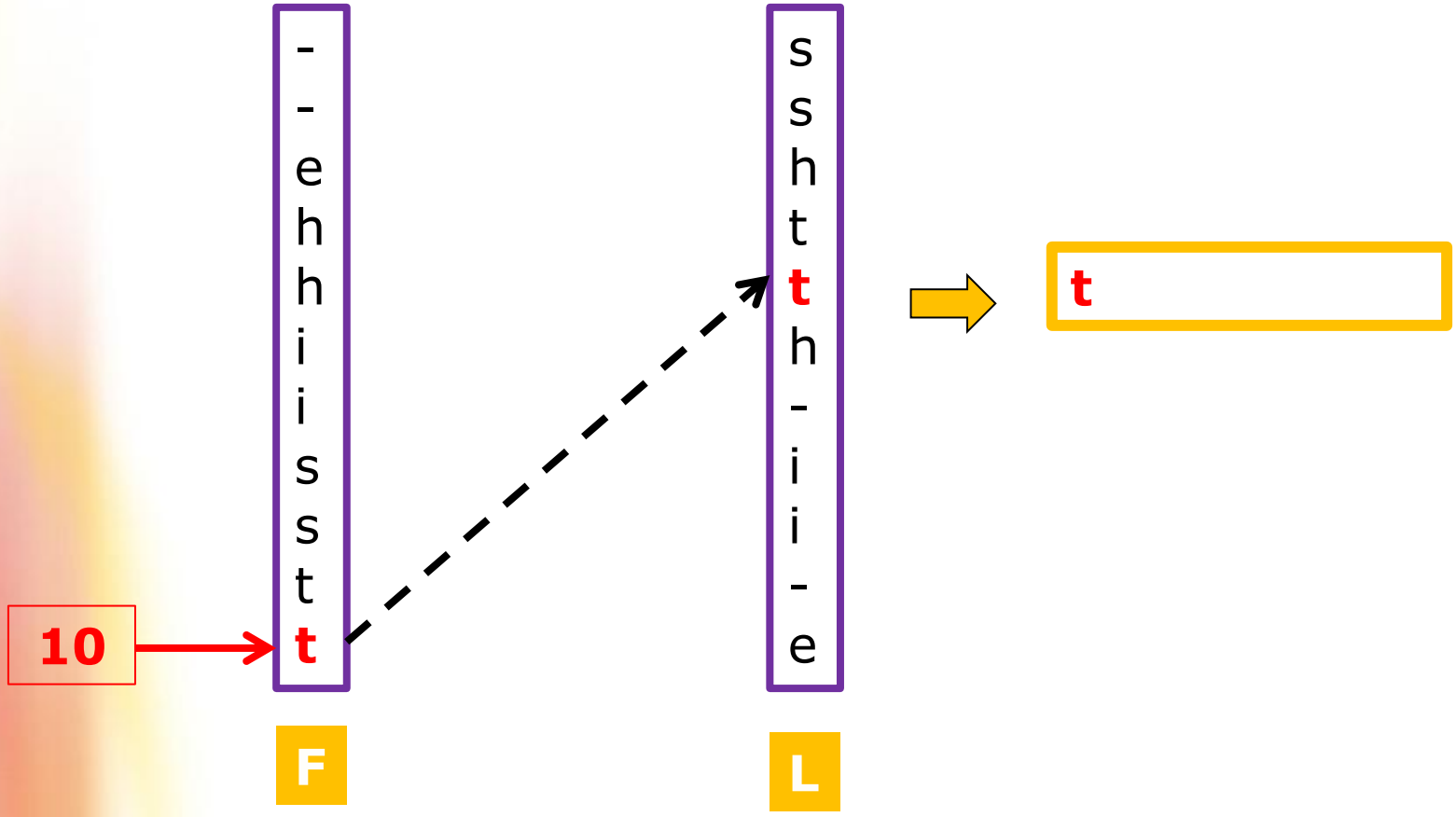
Exemple: Décodage BWT (3)

- Séquence: L= sshtth-ii-e, Indice: 10



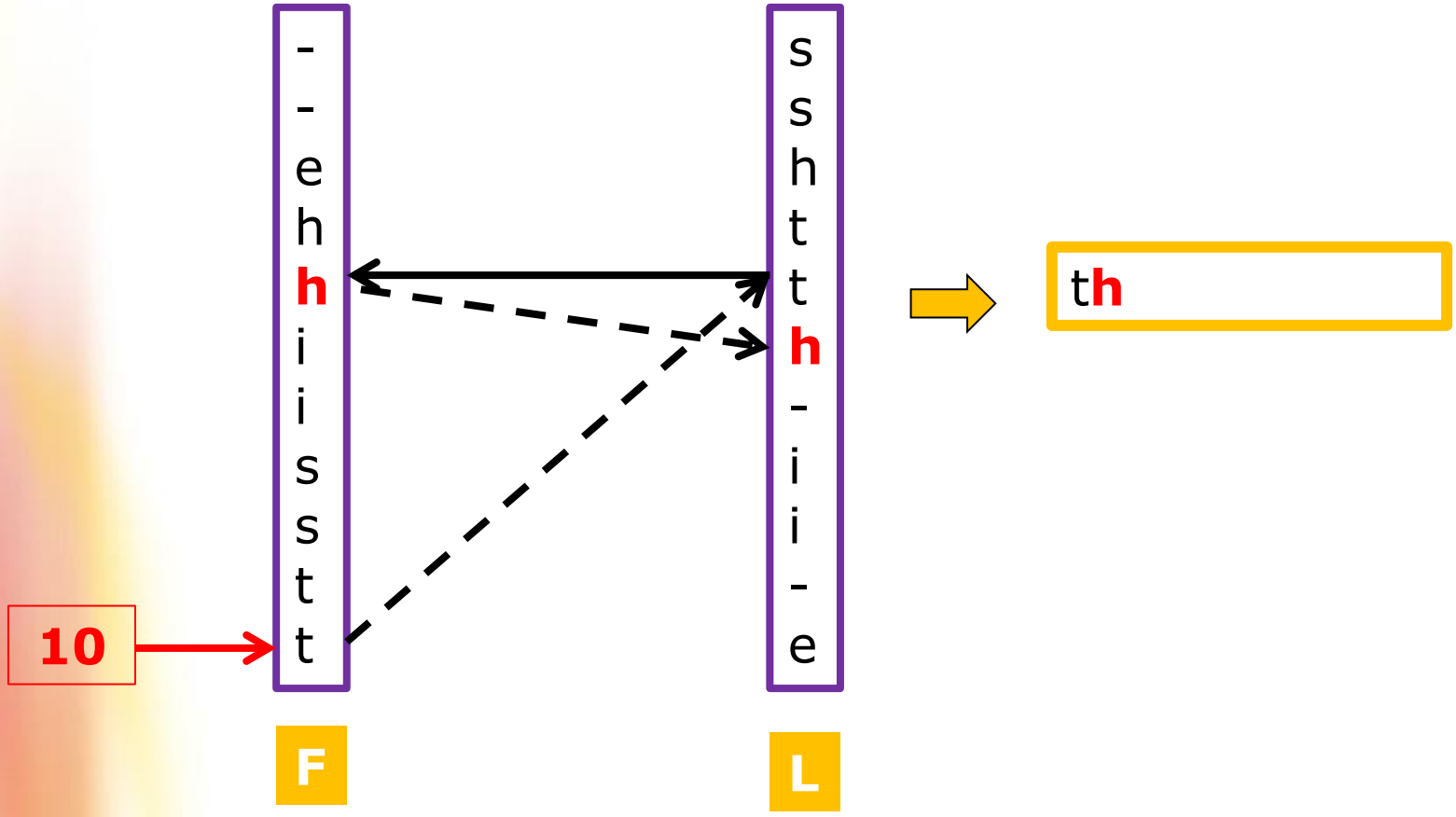
Exemple: Décodage BWT (4) [une autre méthode]

- Séquence: L= sshtth-ii-e, Indice: 10



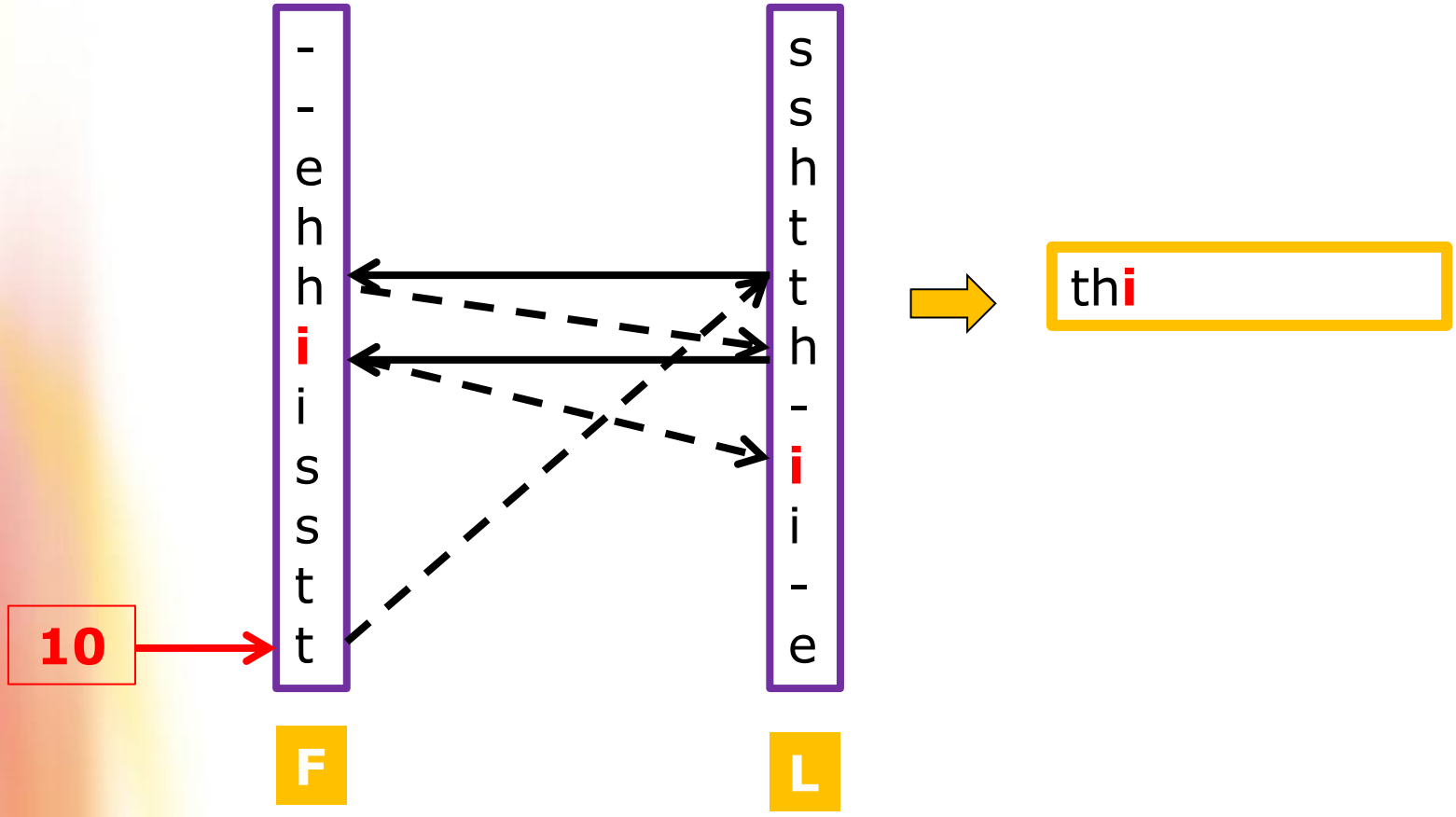
Exemple: Décodage BWT (5) [une autre méthode]

- Séquence: L= sshtth-ii-e, Indice: 10



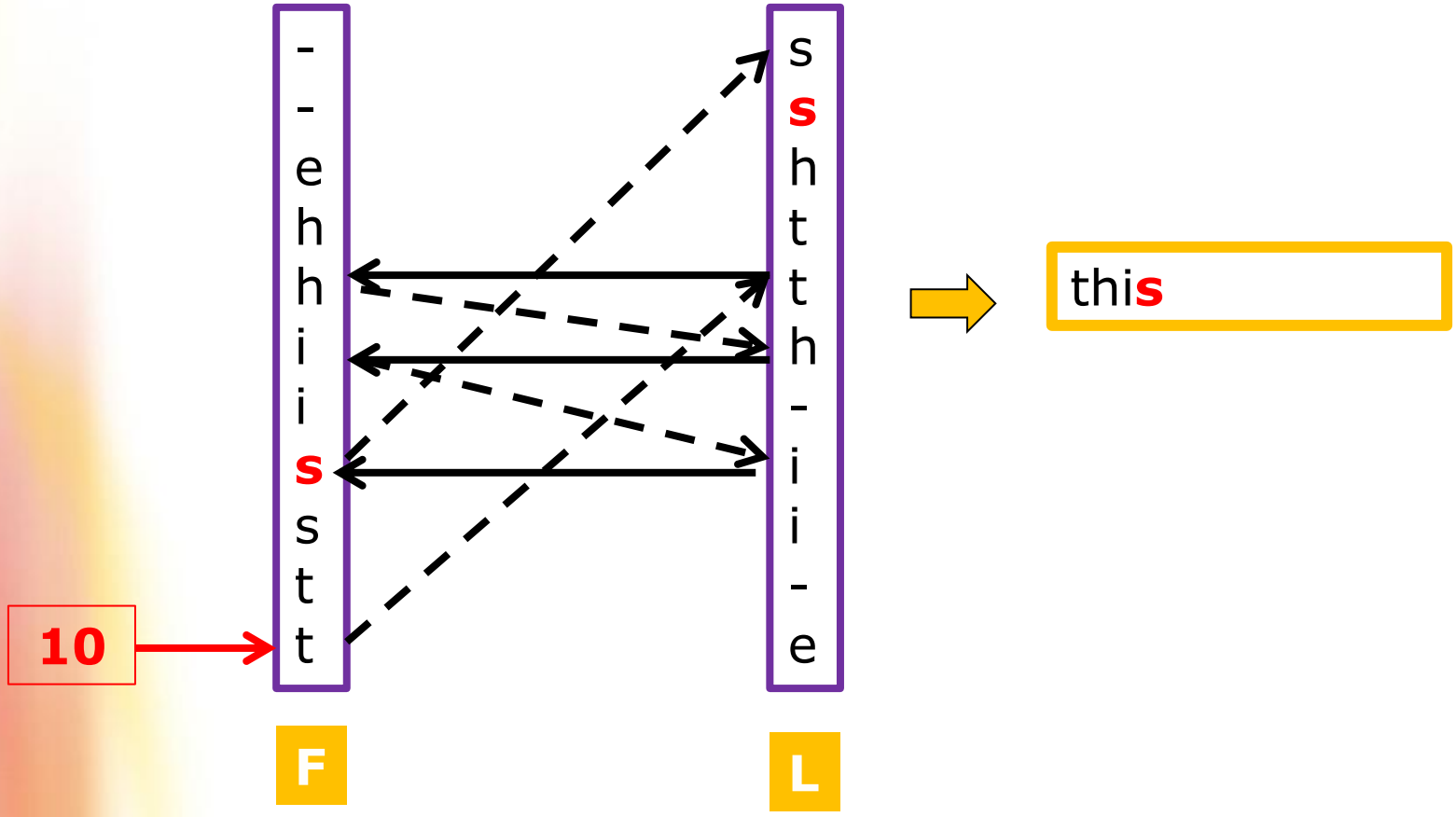
Exemple: Décodage BWT (6) [une autre méthode]

- Séquence: L= sshtth-ii-e, Indice: 10



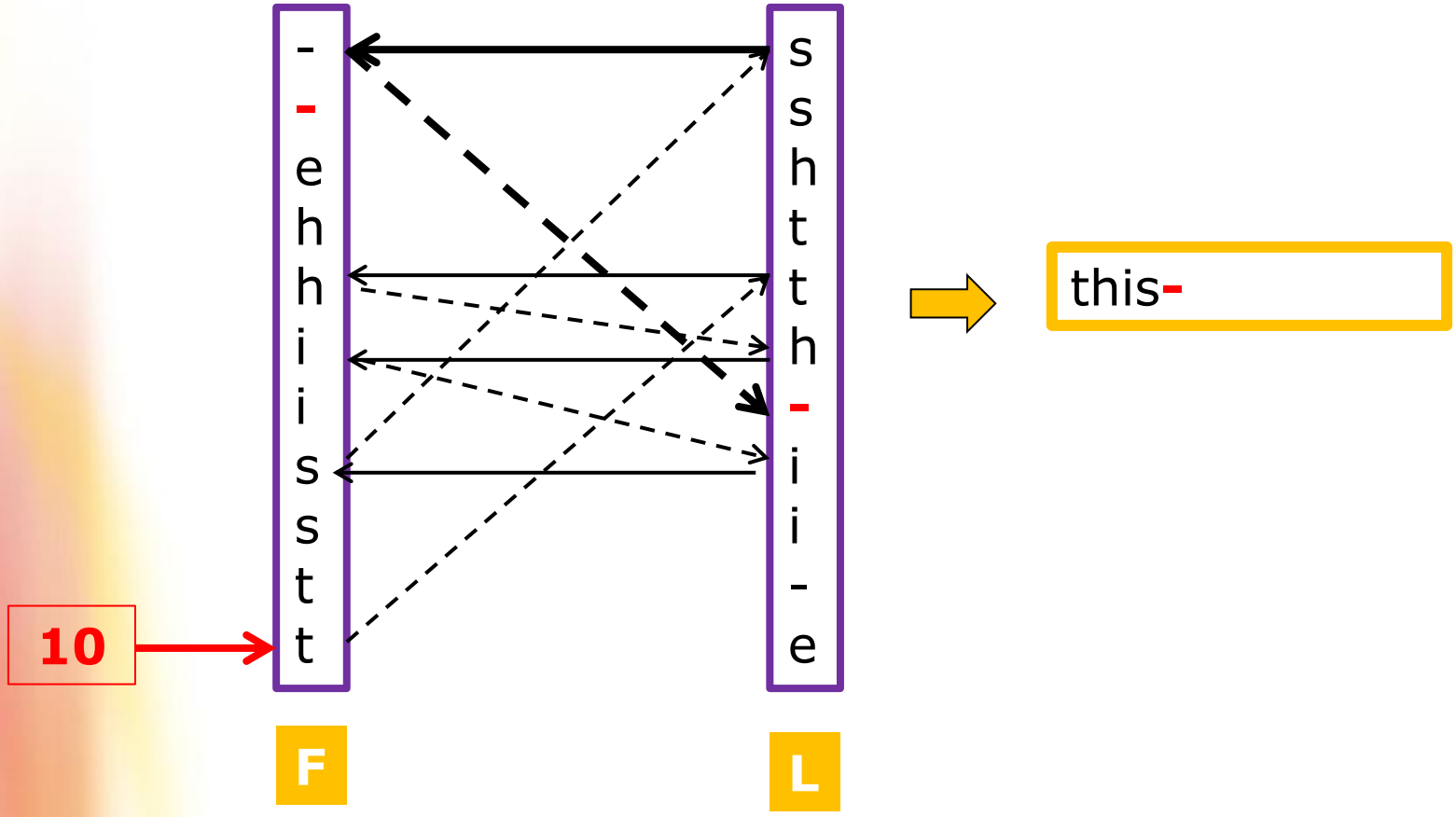
Exemple: Décodage BWT (7) [une autre méthode]

- Séquence: L= sshtth-ii-e, Indice: 10



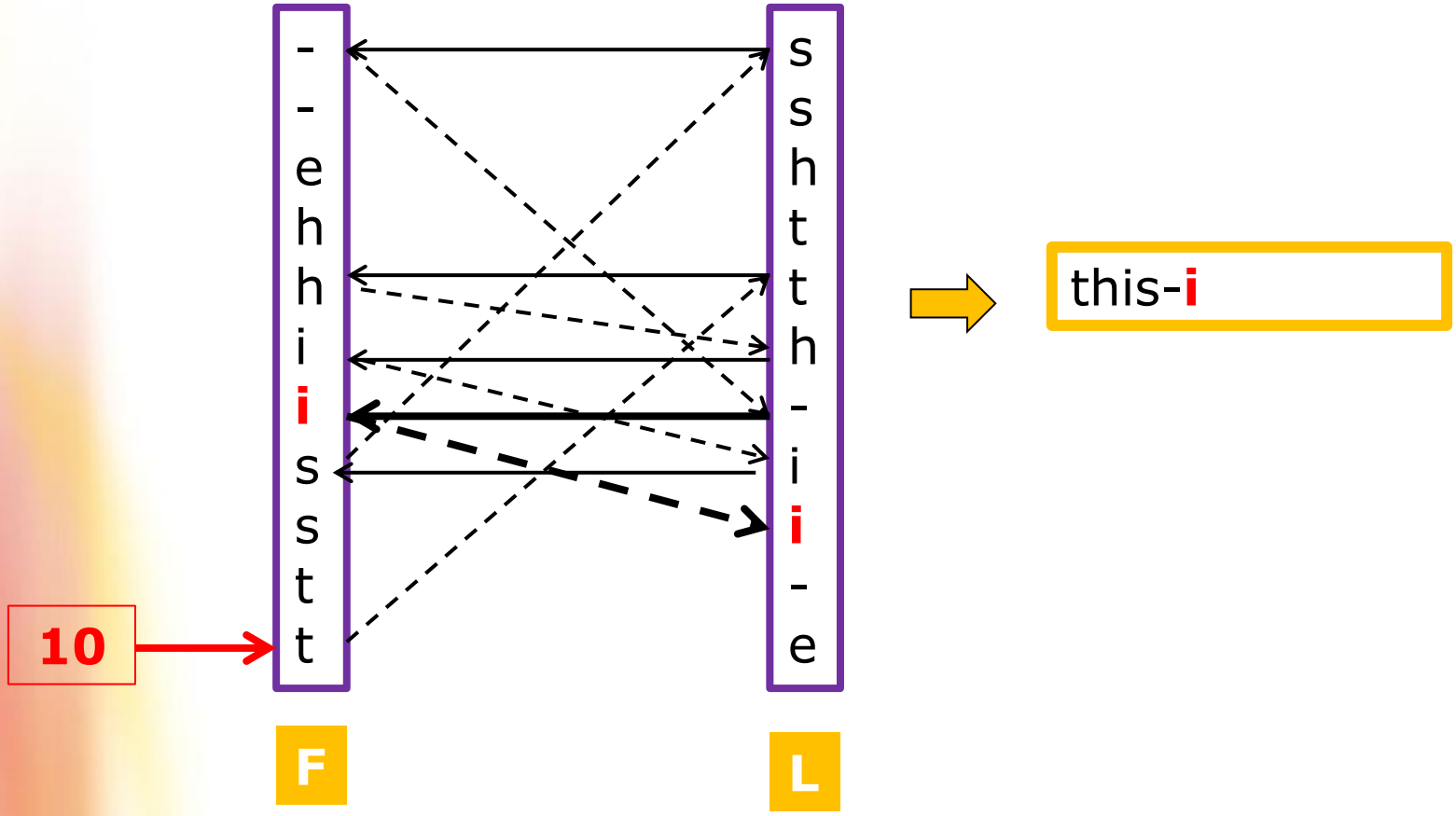
Exemple: Décodage BWT (8) [une autre méthode]

- Séquence: L= sshtth-ii-e, Indice: 10



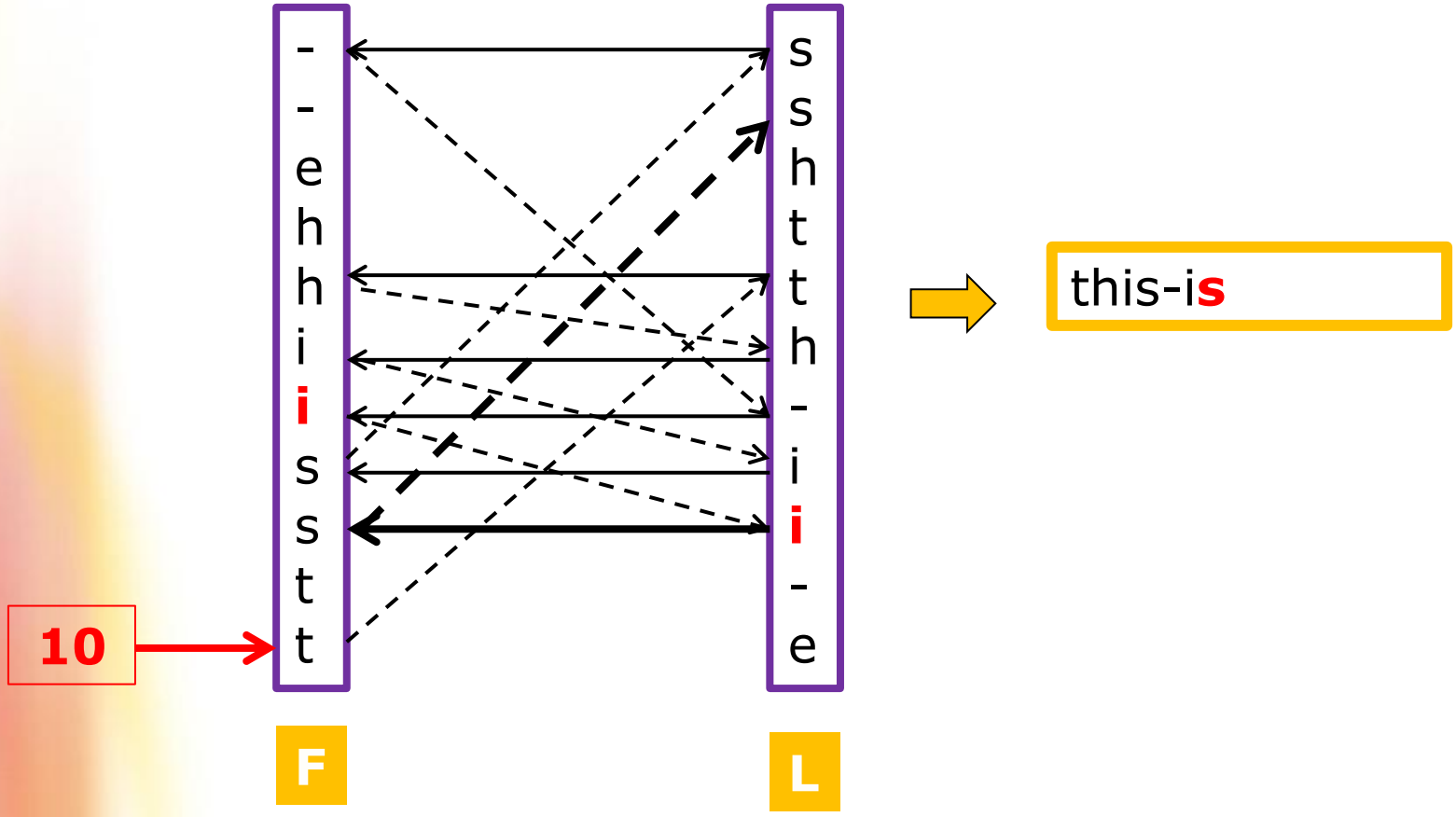
Exemple: Décodage BWT (9) [une autre méthode]

- Séquence: L= sshtth-ii-e, Indice: 10



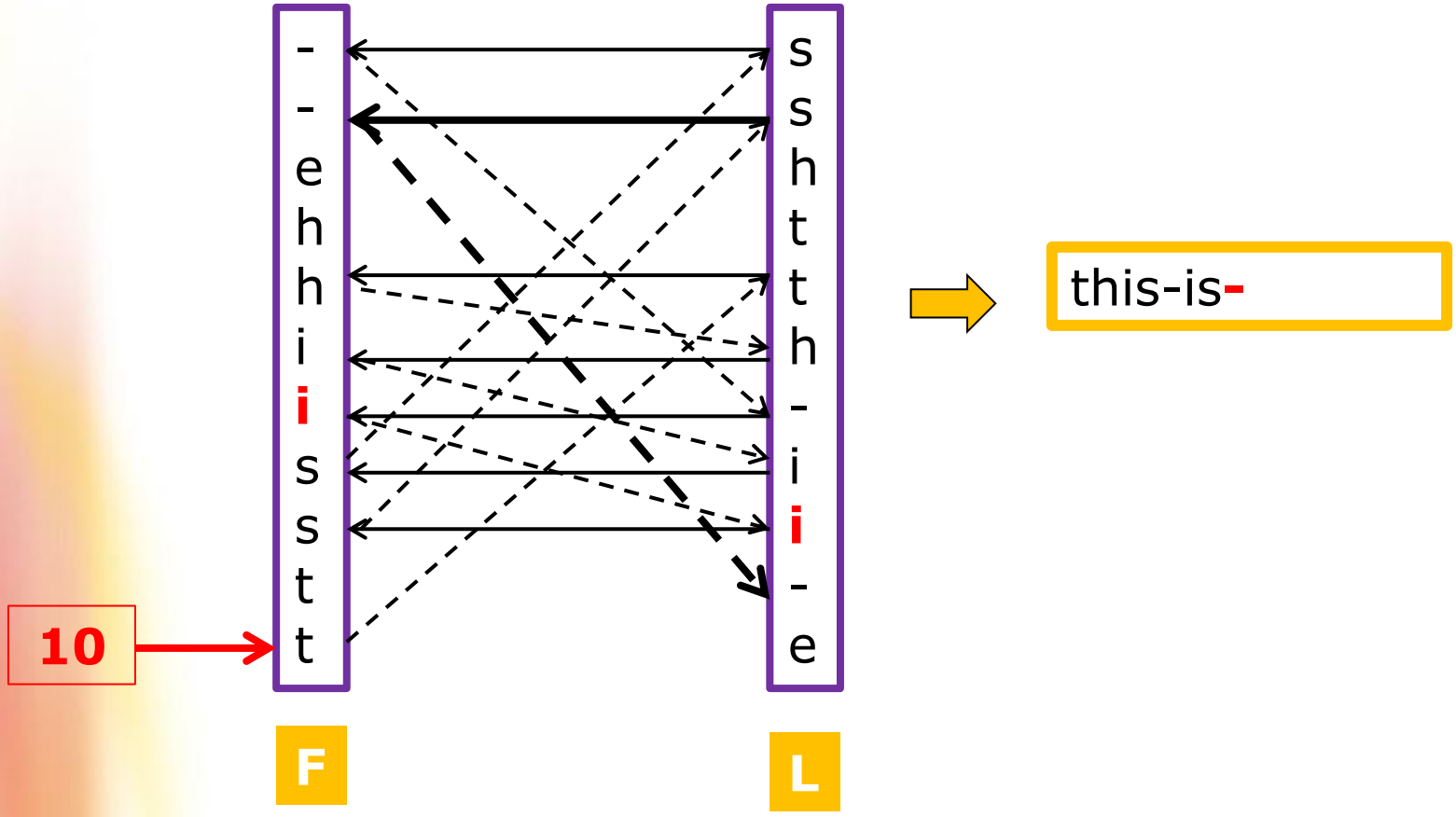
Exemple: Décodage BWT (10) [une autre méthode]

- Séquence: L= sshtth-ii-e, Indice: 10



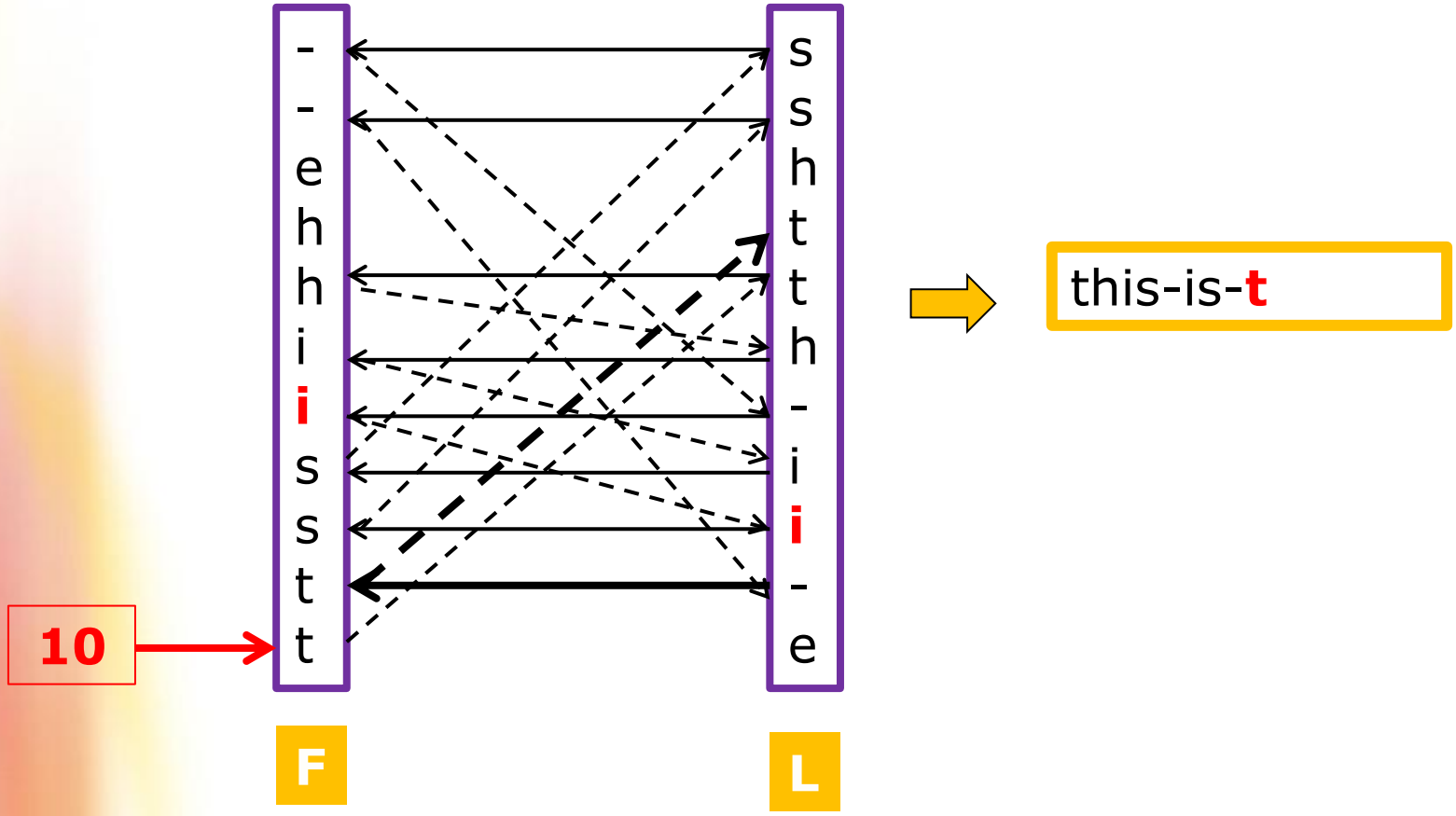
Exemple: Décodage BWT (10) [une autre méthode]

- Séquence: L= sshtth-ii-e, Indice: 10



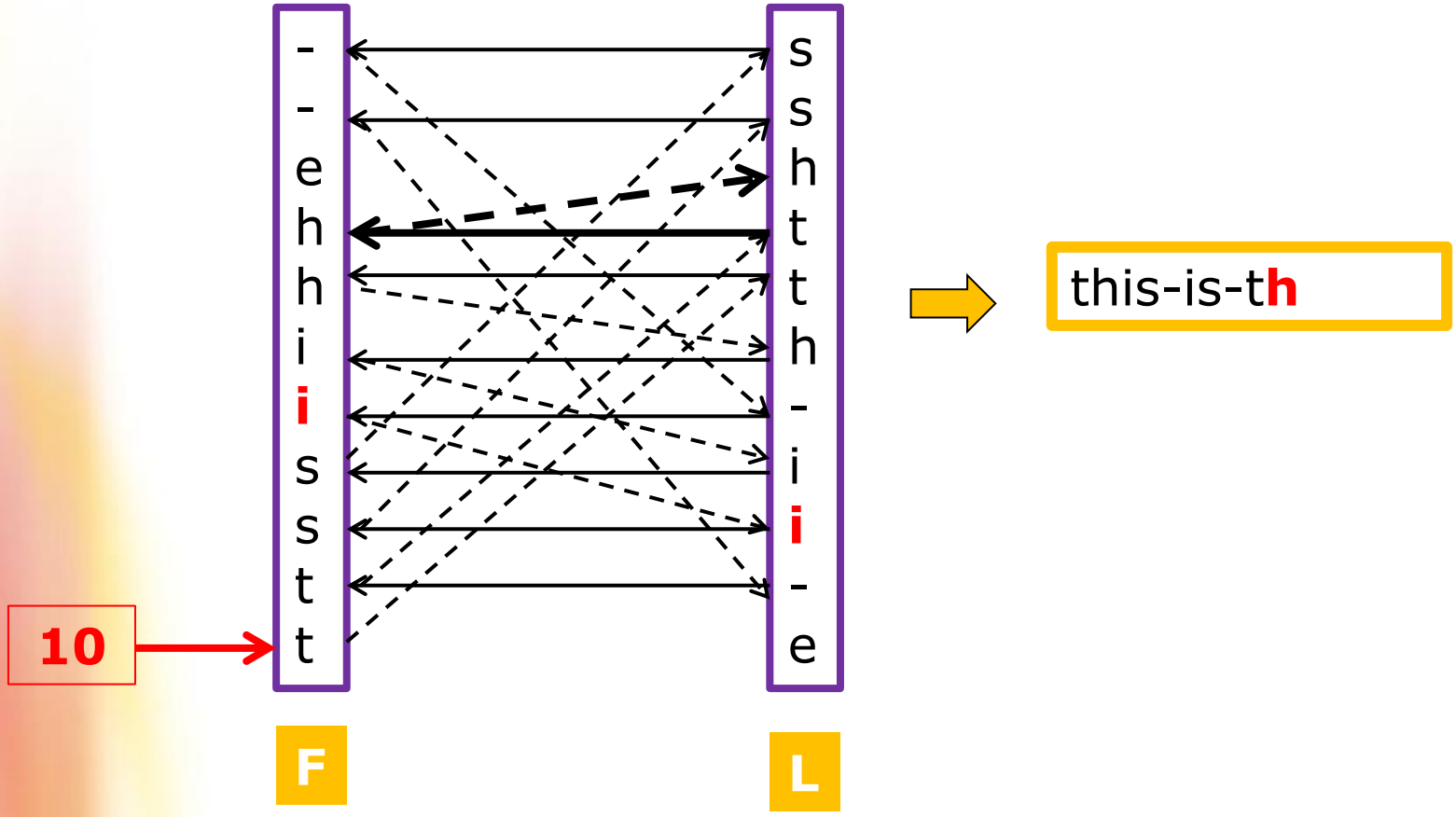
Exemple: Décodage BWT (11) [une autre méthode]

- Séquence: L= sshtth-ii-e, Indice: 10



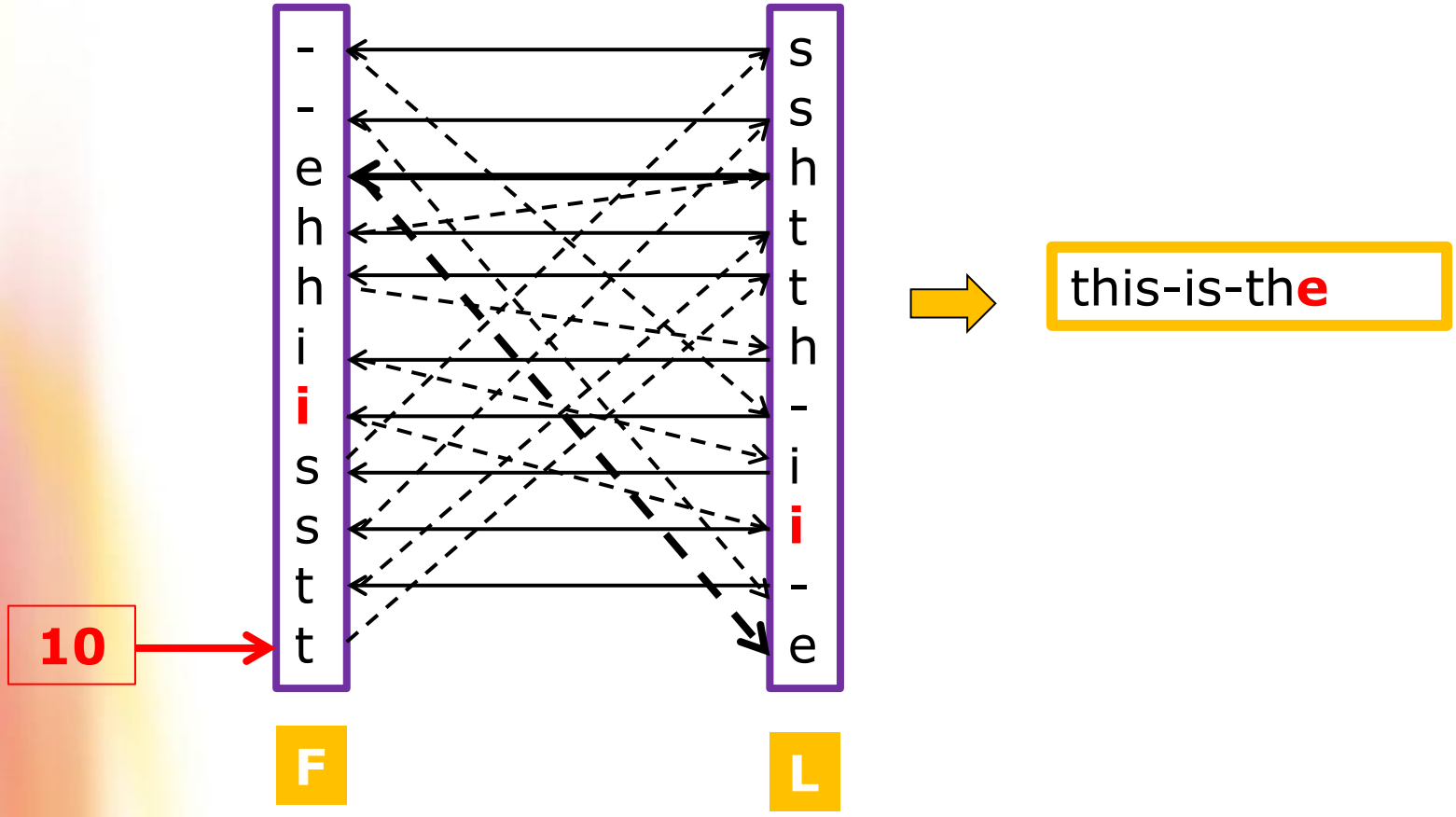
Exemple: Décodage BWT (12) [une autre méthode]

- Séquence: L= sshtth-ii-e, Indice: 10



Exemple: Décodage BWT (13) [une autre méthode]

- Séquence: L= sshtth-ii-e, Indice: 10



Codage MTF: Move-to-Front

- Observation
  - La transformation BWT donne une séquence avec une structure de lettres identiques qui se répètent successivement.
  - Pour bénéficier de cette structure on utilise le codage MTF.
  - On commence par un tri initial de l'alphabet de la source.
  - On assigne '0' au symbole au haut de la liste.
  - Le suivant '1' et ainsi de suite ('2' pour le suivant, '3' …).
  - Nouvelle lettre → envoi du nombre correspondant.

    → Déplacement de la lettre au top de la liste.

→ Ainsi une série de lettres identiques consécutives se traduit par une série de '0'

Exemple de codage MTF (1)

- Observation
  - L= **s**shtth-ii-e
  - A={-,e,h,i,s,t}
  - Arrangement initial

0	1	2	3	**4**	5
-	e	h	i	**s**	t

  - Séquence à envoyer: **4**
  - Mise à jour de la table

0	1	2	3	4	5
s	-	e	h	i	t

Exemple de codage MTF (2)

- Observation
  - L= s**s**htth-ii-e

0	1	2	3	4	5
s	-	e	h	i	t

  - Séquence à envoyer: 4 **0**
  - Mise à jour de la table

0	1	2	3	4	5
s	-	e	h	i	t

Exemple de codage MTF (3)

- Observation
  - L= ss**h**tth-ii-e

0	1	2	**3**	4	5
s	-	e	**h**	i	t

  - Séquence à envoyer: 4 0 **3**
  - Mise à jour de la table

0	1	2	3	4	5
h	s	-	e	i	t

Exemple de codage MTF (4)

- Observation
  - L= ssh**t**th-ii-e

0	1	2	3	4	**5**
h	s	-	e	i	**t**

  - Séquence à envoyer: 4 0 3 **5**
  - Mise à jour de la table

0	1	2	3	4	5
t	h	s	-	e	i

Exemple de codage MTF (5)

- Observation
  - L= ssht**t**h-ii-e

0	1	2	3	4	5
t	h	s	-	e	i

  - Séquence à envoyer: 4 0 3 5 **0**
  - Mise à jour de la table

0	1	2	3	4	5
t	h	s	-	e	i

Exemple de codage MTF (6)

- Observation
  - L= ssht**t**h-ii-e

0	**1**	2	3	4	5
t	**h**	s	-	e	i

  - Séquence à envoyer: 4 0 3 5 0 **1**
  - Mise à jour de la table

0	1	2	3	4	5
h	t	s	-	e	i

Exemple de codage MTF (7)

- Observation
  - L= sshtth-ii-e
  - Séquence à envoyer: 4 0 3 5 0 1 3 5 0 1 5
  - Le résultat ne montre pas beaucoup de '0' car l'exemple traite une séquence courte.
  - Mais si on considère un fichier texte de grande taille on va avoir plus de séquence de '0' et des valeurs généralement faible.
  - Comme les faibles valeurs sont plus fréquentes on peut utiliser un code qui assigne de petits code aux valeurs faibles.
  - → **Huffman ou codage arithmétique.**