

TP RÉSEAUX

LE CRC - CODE CYCLIQUE DE REDONDANCE

Ballot Corentin, Simon Camille

09/04/2017

Ce compte-rendu présente l'implémentation d'un programme JAVA permettant de coder et de vérifier une chaîne binaire à l'aide de la méthode CRC.

L'archive Reseau-TP3-CorentinBallot-CamilleSimon.tar.gz. contient :

- Les fichiers JAVA permettant l'exécution du programme
- Le présent rapport

Le sujet du TP est accessible ici : [TP Réseaux - Les Codes Cycliques de Redondance](#).

1 Analyse du sujet

Dans ce TP, nous devons réaliser un programme qui code et décode une chaîne binaire. La méthode utilisée est celle des Codes Cycliques de Redondance abrégée sous le sigle CRC. Cette méthode consiste à utiliser un *polynôme générateur* pour coder et décoder un message.

1.1 Principe du code CRC

Le code CRC utilise une représentation polynomiale des chaînes de bits. Une chaîne $m_1m_2...m_{n-1}m_n$ peut être représentée sous la forme polynomiale $m_n + m_{n-1}x + ...m_2x^{n-2} + m_1x^{n-1}$. Par exemple, la chaîne 1100101 a pour représentation polynomiale : $1x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 0x^1 + 1x^0 = x^6 + x^5 + x^2 + 1$.

1.2 Génération du code CRC

Pour générer un code CRC, on commence par choisir un polynôme générateur, puis, on le transforme en chaîne binaire. Il faut ensuite ajouter au message un nombre de zéro égal au degré du polynôme générateur. Pour le polynôme $x^4 + x^2 + x$ de degré 4 et le message 11100111, on obtient la chaîne 111001110000. On soustrait à la chaîne obtenue la chaîne du polynôme générateur, on réitère l'opération jusqu'à ce que le reste de la soustraction soit d'un nombre de bits inférieur au nombre de bits de la chaîne du polynôme générateur.

Voici une illustration avec le message du paragraphe précédent:

1	1	1	0	0	1	1	1	0	0	0	0
1	0	1	1	0							
<hr/>											
0	1	0	1	0	1	1	1	0	0	0	0
	1	0	1	1	0						
<hr/>											
	0	0	0	1	1	1	1	0	0	0	0
				1	0	1	1	0			
<hr/>											
				0	1	0	0	0	0	0	0
					1	0	1	1	0		
<hr/>											
					0	0	1	1	0	0	0
							1	0	1	1	0
<hr/>											
							0	1	1	1	0

Un fois les soustractions effectuées, il ne reste plus qu'à ajouter le code CRC au message à envoyer. Dans notre cas, le message envoyé sera 11100111 1110.

1.3 Vérification d'un message CRC

Le récepteur du message doit vérifier que celui-ci ne comporte pas d'erreur. Pour ce faire, on soustrait la chaîne obtenue grâce au polynôme générateur au message reçu. On réitère la soustraction jusqu'à ce que le degré du résultat de la soustraction soit inférieur au degré du polynôme générateur. À la fin du processus, si le reste est nul alors le message reçu est correct. Sinon, c'est qu'il y a une erreur.

Voici la vérification du mot 111001111110 par la chaîne du polynôme $x^4 + x^2 + x$:

1	1	1	0	0	1	1	1	1	1	1	0
1	0	1	1	0							
<hr/>											
0	1	0	1	0	1	1	1	1	1	1	0
	1	0	1	1	0						
<hr/>											
	0	0	0	1	1	1	1	1	1	1	0
				1	0	1	1	0			
<hr/>											
				0	1	0	0	1	1	1	0
					1	0	1	1	0		
<hr/>											
					0	0	1	0	1	1	0
							1	0	1	1	0
<hr/>											
							0	0	0	0	0

2 Choix techniques & mise en œuvre

2.1 Choix techniques

Dans ce TP nous devons réaliser une interface graphique. Nous avons choisi d'utiliser Swing. Bien que cette technologie soit ancienne, elle reste largement utilisée et enseignée.

2.2 Mise en œuvre

Le travail demandé peut être séparé en deux volets, d'une part la conception des algorithmes de codage et de décodage, et d'autre part, l'aspect graphique de l'application.

2.2.1 Algorithmes

Les algorithmes permettant la génération et la vérification des mots obtenues grâce à la méthode CRC se trouvent dans le fichier *Resultat.java*. Nous allons maintenant décrire le fonctionnement de ces deux algorithmes :

- CODAGE

On commence par déterminer le nombre de zéros à ajouter au message. On ajoute au message autant de zéro que le degré du polynôme générateur. Tant que le degré du résultat est supérieur ou égal au degré du polynôme générateur, on effectue la soustraction binaire du résultat par le polynôme générateur.

Enfin, on ajoute des zéros pour que le résultat contienne *degré du polynôme - 1* bits, puis on transmet le résultat à la fonction d'affichage.

- CORRECTION

On commence par effectuer des soustractions successives tant que le degré du résultat est supérieur ou égal au degré du polynôme générateur.

Si la chaîne est vide, c'est que le message est bon, on affiche alors "OK" à l'utilisateur. Sinon, on affiche le message "ERREUR" à l'utilisateur ainsi que le résultat des soustractions successives.

2.2.2 Design

Une part non négligeable du travail consiste à créer une interface utilisateur simple, efficace et intuitive. Voici à quoi ressemble notre application :

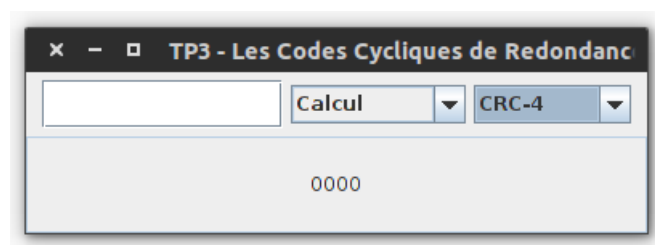


Figure 1: Visuel de l'application

La fenêtre est séparée en deux parties, la partie supérieure (1) où l'utilisateur va entrer ses informations et la partie inférieure cf. **Figure: 3** où le programme va afficher le résultat. L'ensemble de la fenêtre est générée par le fichier *Fenetre.java*.

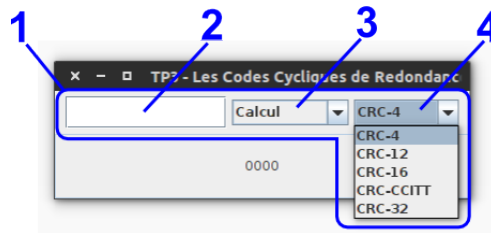


Figure 2: Visuel de l'application. (1) L'espace de saisie des paramètres, (2) JTextField pour le message, (3) JComboBox de sélection de l'opération à effectuer, (4) JComboBox de sélection du polynôme générateur.

La partie inférieure va être différente en fonction de ce qui est demandé au programme. Pour le calcul, il affichera le mot avec suivi du code CRC (5). Pour la vérification, on aura soit "OK" si le message ne contient pas d'erreur (6), soit, il affichera "ERREUR" ainsi que le reste de la division (7).

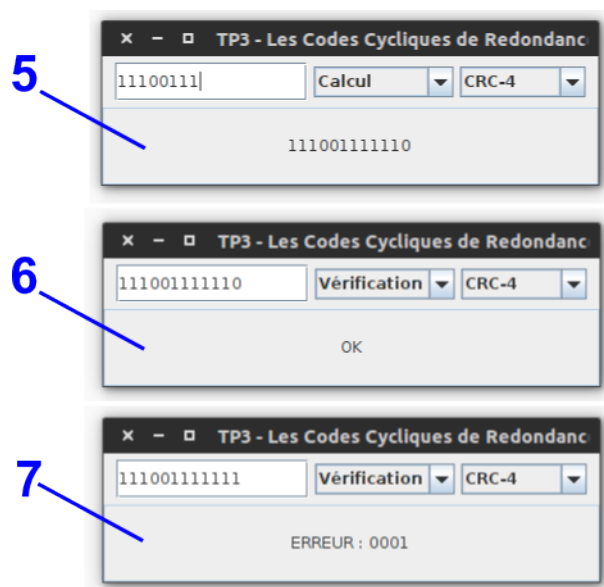


Figure 3: Visuel de l'application. (5) Message après codage, (6) le message ne contient pas d'erreur, (7) le message contient une erreur.

3 Tests

Voici les résultat de l'application pour le codage et la vérification :

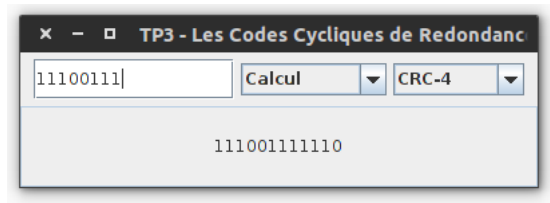


Figure 4: Résultat de l'application pour le codage du message 11100111

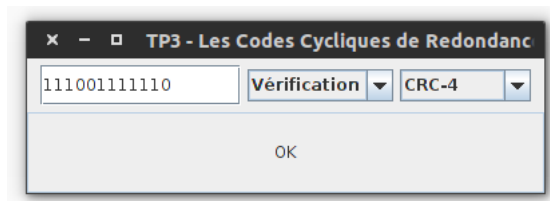


Figure 5: Résultat de l'application pour la vérification d'un message correct

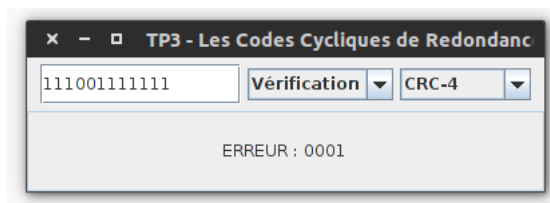


Figure 6: Résultat de l'application pour la vérification d'un message incorrect

4 Conclusion

Dans ce TP, nous avons réalisé une application graphique permettant de coder et de vérifier un message à l'aide de la méthode CRC. Nous avons appliqué la méthode vue en TD. Ce travail nous a permis non seulement d'exercer nos compétences en algorithmique mais encore de comprendre de manière approfondie le fonctionnement de la méthode CRC.