Alexandria University,

Faculty of Engineering,

Computer and Systems Engineering Dept.

Digital Systems 2

CS232

*Lab 4*

Uniquely Decipherable Codes

Names:

1. Ahmed El-Sayed Molahez (3)

2. Ziad Hisham Ali (19)

3. Abdelrahman El-Sayed Ahmed (22)

1. Youssef Ahmed Sayed Gabriel (61)
2. Problem Statement:

Design a Ciphering/Deciphering machine and try it out to send & receive messages (e.g., try out the given example above, and any other example with 10 symbols)

1. Modules :

* Cipher “Class”

It’s responsible for sending the ciphered message bit by bit using the user input through a push button then send it to the other machine which after receiving the whole message starts deciphering it.

* Decipher “Class”

It’s responsible for deciphering the received message, the process of receiving the bit by bit message from the cipher machine is as follows:

1. It awaits for the user to send a high signal to start receiving the message.
2. The user is given 2 seconds to enter the bit value either high for one or low for zero.
3. The user is given another 2 seconds to finish the message by sending a high signal.
4. If no high signal is received, then step 1 is repeated.
5. Space character :

Add an extra symbol to represent the “SPACE” to your code, and make sure it is still uniquely decipherable.

We’ve tried several code words assignment to the space symbol but we couldn’t find such code word that results in a uniquely decipherable code.

Follows are some examples of code words for space symbol (represented as F) that fails in the uniquely decipherability test.

Test unique decipherability of {11, 011, 001, 01, 00, 10} for

source alphabet {A, B, C, D, E, F}

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
| S | (B1C1)(B1D1)(B1E1)(C1D1)(C1E1)(D1E1) | (A1F1) |
| B1C1 | - | - |
| B1D1 | - | (SB2) |
| B1E1 | - | - |
| C1D1 | - | - |
| C1E1 | (SC2) | - |
| D1E1 | - | - |
| A1F1 | - | - |
| SB2 | - | (SA1)(SF1) |
| SC2 | - | (SA1)(SF1) |
| SA1 | - | (SA1)(SF1) |
| SF1 | (SB1)(SC1)(SD1)(SE1) | - |
| SB1 | - | (A1B2)(B2F1) |
| SC1 | (B1C2)(C1C2)(C2D1)(C2E1) | - |
| SD1 | - | (SA1)(SF1) |
| SE1 | - | - |
| A1B2 | - | (SS) |
| B2F1 | - | - |
| B1C2 | - | (SB2) |
| C1C2 | - | - |
| C2D1 | - | (SS) |
| C2E1 | - | - |

A → **S** 1 **A1** 1 **S**

B → **S** 0 **B1** 1 **B2** 1 **S**

C → **S** 0 **C1** 0 **C2** 1 **S**

D → **S** 0 **D1** 1 **S**

E → **S** 0 **E1** 0 **S**

F → **S** 1 **F1** 0 **S**

Test unique decipherability of {11, 011, 001, 01, 00, 100} for

source alphabet {A, B, C, D, E, F}

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
| S | (B1C1)(B1D1)(B1E1)(C1D1)(C1E1)(D1E1) | (A1F1) |
| B1C1 | - | - |
| B1D1 | - | (SB2) |
| B1E1 | - | - |
| C1D1 | - | - |
| C1E1 | (SC2) | - |
| D1E1 | - | - |
| A1F1 | - | - |
| SB2 | - | (SA1)(SF1) |
| SC2 | - | (SA1)(SF1) |
| SA1 | - | (SA1)(SF1) |
| SF1 | (SB1)(SC1)(SD1)(SE1) | - |
| SB1 | - | (A1B2)(B2F1) |
| SC1 | (B1C2)(C1C2)(C2D1)(C2E1) | - |
| SD1 | - | (SA1)(SF1) |
| SE1 | - | - |
| A1B2 | - | (SS) |
| B2F1 | - | - |
| B1C2 | - | (SB2) |
| C1C2 | - | - |
| C2D1 | - | (SS) |
| C2E1 | - | - |

A → **S** 1 **A1** 1 **S**

B → **S** 0 **B1** 1 **B2** 1 **S**

C → **S** 0 **C1** 0 **C2** 1 **S**

D → **S** 0 **D1** 1 **S**

E → **S** 0 **E1** 0 **S**

F → **S** 1 **F1** 0 **F2** 0 **S**

Test unique decipherability of {11, 011, 001, 01, 00, 111} for

source alphabet {A, B, C, D, E, F}

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
| S | (B1C1)(B1D1)(B1E1)(C1D1)(C1E1)(D1E1) | (A1F1) |
| B1C1 | - | - |
| B1D1 | - | (SB2) |
| B1E1 | - | - |
| C1D1 | - | - |
| C1E1 | (SC2) | - |
| D1E1 | - | - |
| A1F1 | - | (SF2) |
| SB2 | - | (SA1)(SF1) |
| SC2 | - | (SA1)(SF1) |
| SF2 | - | (SA1)(SF1) |
| SA1 | - | (SA1)(SF1) |
| SF1 | - | (A1F2)(F1F2) |
| A1F2 | - | (SS) |
| F1F2 | - | (SF2) |

A → **S** 1 **A1** 1 **S**

B → **S** 0 **B1** 1 **B2** 1 **S**

C → **S** 0 **C1** 0 **C2** 1 **S**

D → **S** 0 **D1** 1 **S**

E → **S** 0 **E1** 0 **S**

F → **S** 1 **F1** 1 **F2** 1 **S**

1. “C” character :

Set Code Word of character “C” to 101 and check if it’s Uniquely Decipherable Code.

We’ve tested it using our code using the message “CABEBDEA” and it wasn’t uniquely decipherable and follows the test using testing table:

Test unique decipherability of {11, 011, 101, 01, 00} for

source alphabet {A, B, C, D, E}

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
| S | (B1D1)(B1E1)(D1E1) | (A1C1) |
| B1D1 | - | (SB2) |
| B1E1 | - | - |
| D1E1 | - | - |
| A1C1 | - | - |
| SB2 | - | (SA1)(SC1) |
| SA1 | - | (SA1)(SC1) |
| SC1 | (B1C2)(C2D1)(C2E1) | - |
| B1C2 | - | (SB2) |
| C2D1 | - | (SS) |
| C2E1 | - | - |

A → **S** 1 **A1** 1 **S**

B → **S** 0 **B1** 1 **B2** 1 **S**

C → **S** 1 **C1** 0 **C2** 1 **S**

D → **S** 0 **D1** 1 **S**

E → **S** 0 **E1** 0 **S**