



Codificação de Canal: Correção de erro por codificação convolucional.



Departamento de Eletrônica e Computação

Centro de Tecnologia

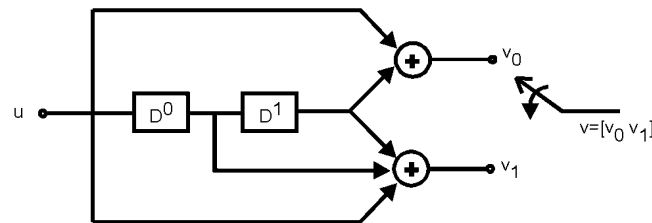
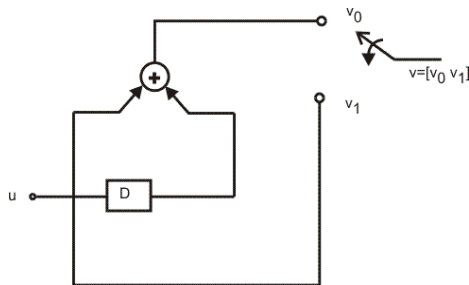
ELC1120 – TELECOMUNICAÇÕES II

Profa. Candice Müller Prof. Fernando DeCastro

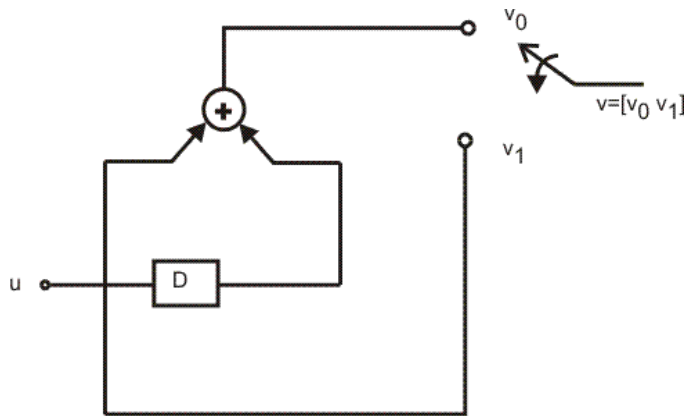
Códigos Convolucionais – Decodificador de Viterbi

- Um código convolucional é gerado pela combinação linear em $\mathbf{GF}(2)$ das saídas de um *shift-register* de K estágios.
- A sequência de bits a ser codificada é aplicada na entrada do *shift-register*, e este executa a convolução em $\mathbf{GF}(2)$ entre a sequência de entrada e a resposta ao impulso da máquina de estado (state machine) representada pelo *shift-register*.
- A saída da máquina de estado constitui, portanto, a sequência codificada.
- Diferem dos códigos de bloco porque o mapeamento entrada/saída do codificador não é dado por uma matriz geradora e, sim, dependente do estado da máquina.

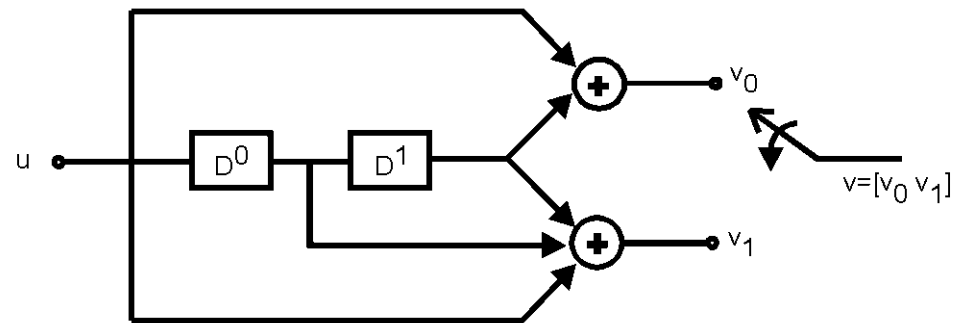
Exemplos de Codificadores Convolucionais



- O n.º. de estados da máquina de estado de um codificador convolucional é 2^K , sendo K o n.º. de estágios do *shift-register*.
- No contexto de códigos convolucionais $(K + 1)$ recebe o nome de *constraint length*.
- A razão entre o n.º. de entradas e o n.º. de saídas da máquina de estados define a razão de codificação R_c do codificador.



Codificador Convolucional com $K = 1$ estágio
($2^K = 2$ estados) e $R_c = 1/2$.



Codificador Convolucional com $K = 2$ estágios
($2^K = 4$ estados) e $R_c = 1/2$.

Um codificador Convolutacional de 1 estágio, 2 estados e $R_c = 1/2$ poderá apresentar, por exemplo, o seguinte mapeamento entrada/saída →

Entrada	Estado	Saída
0	0	01
1	0	10
0	1	11
1	1	00

Se o mapeamento entrada/saída do codificador não é dado por uma matriz geradora e, sim, dependente do estado da máquina, uma entrada “1”, como no exemplo acima, poderá ser mapeada em uma saída “10”, mas também poderá ser mapeada em uma saída “00”.

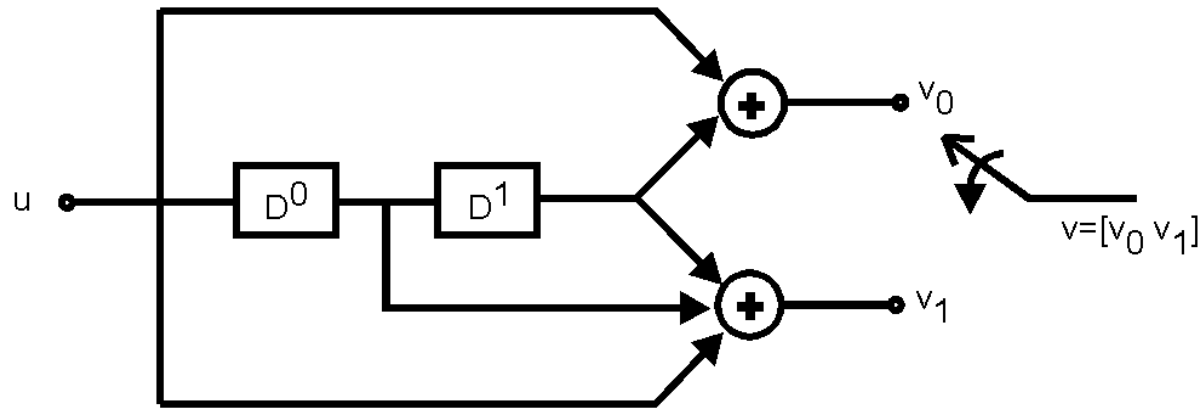
Por que este codificador será utilizável?

Como uma máquina de estados construída a partir de um shift-register apresenta um conjunto finito de transições permitidas entre estados, quando a sequência a ser codificada é a ela submetida, implicitamente ficarão restritas as transições da sequência codificada em sua saída.

Ou seja, há um conjunto conhecido e finito de possíveis transições de estado.

Se o receptor conhecer a tabela de transições permitidas, então os erros gerados por degradação do sinal no canal de comunicações poderão ser identificados e corrigidos por um Decodificador de Viterbi.

A Figura abaixo apresenta um codificador convolucional com 2 estágios e 4 estados e $R_c = 1/2$.

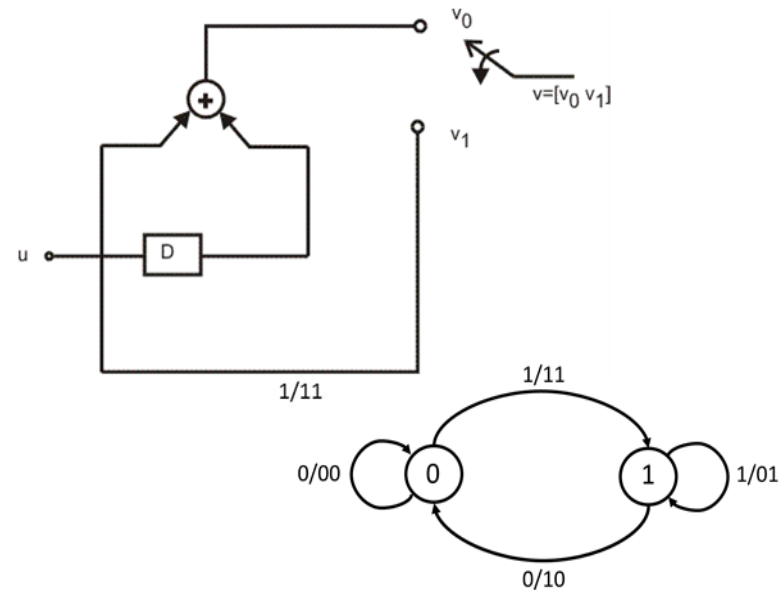


- A sequência de bits a ser codificada é representada por u , e a saída do codificador é a sequência de bits v .
- Visto que $R_c = 1/2$, para cada bit de u são gerados dois bits em v .
- O estágio D^0 transfere o valor lógico em sua entrada para a sua saída **imediatamente após** a ocorrência da borda de descida do pulso de *clock* (não representado na figura).
- De forma idêntica opera o estágio D^1 .
- Representando a saída do estágio D^0 por D^0 e representando a saída do estágio D^1 por D^1 , então o par de bits $D^0 D^1$ identifica um dos estados da máquina de estado.

Estudaremos a codificação e a decodificação no contexto de Códigos Convolucionais, por meio de dois exemplos:

Exemplo 1:

A Figura ao lado apresenta o diagrama de transição de estados de um codificador convolucional com $K = 1$ estágio, $2^K = 2$ estados e $R_c = 1/2$.



No diagrama de transição de estados deste codificador, Figura ao lado, cada círculo representa um estado (D) dentre os 2 possíveis estados.

O diagrama é construído a partir dos estados individuais considerando as **transições permitidas** a partir de cada estado como consequência do valor lógico de u .

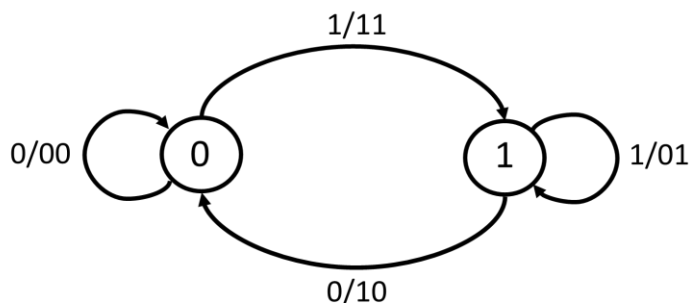
Supondo que a máquina de estado encontre-se no estado **0** (i.e., $D = 0$ na Figura acima).

- Se $u = 1$ a saída resultante é $v = 11$ e, após a borda de descida do *clock*, a máquina vai para o estado **1**.
- Se $u = 0$ a saída resultante é $v = 00$ e, após a borda de descida do *clock*, a máquina vai para o estado **0**.

Supondo que a máquina de estado encontre-se no estado **1** (i.e., $D = 1$ na Figura acima).

- Se $u = 1$ a saída resultante é $v = 01$ e, após a borda de descida do *clock*, a máquina vai para o estado **1**.
- Se $u = 0$ a saída resultante é $v = 10$ e, após a borda de descida do *clock*, a máquina vai para o estado **0**.

Diagrama de Transição de Estados



Exemplo de codificação para o codificador da Figura

u	1	0	1	1	1
D_{atual}	0	1	0	1	1
D_{futuro}	1	0	1	1	1
v	11	10	11	01	01
r	11	10	11	01	01

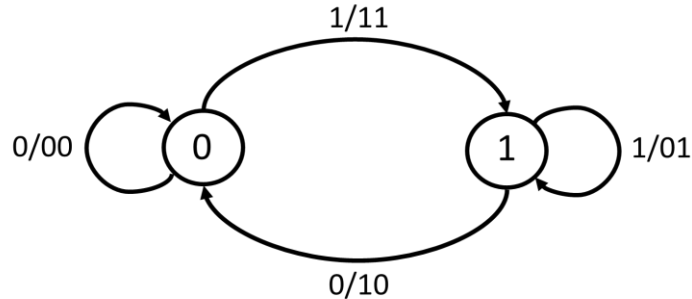
Dada uma seqüência u a ser codificada (1 0 1 1 1), a saída v no codificador de um transmissor digital é enviada ao receptor através do canal de transmissão, sendo recebida como uma seqüência r .

A Tabela acima apresenta uma possível seqüência u e a resultante seqüência v para o codificador da Figura.

É mostrada também a trajetória do estado D à medida que u é codificada, partindo inicialmente do estado 0.

Se nenhuma degradação de sinal ocorreu no canal de transmissão, $r = v$.

Diagrama de Transição de Estados



Exemplo de codificação para o codificador da Figura

u	1	0	1	1	1
D _{atual}	0	1	0	1	1
D _{futuro}	1	0	1	1	1
v	11	10	11	01	01
r	11	10	11	11	01

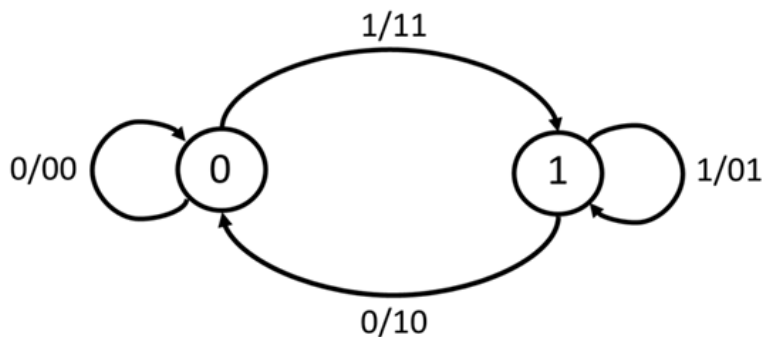


Assumindo que v seja enviado através de um canal de transmissão com ruído/interferência, a Tabela mostra uma possível sequência r recebida com 1 erro.

No receptor digital, o decodificador utiliza um algoritmo de decodificação baseado no princípio de mínima distância (MLSE – *maximum likelihood sequence detector*) denominado **Algoritmo de Viterbi**.

Vamos decodificar a sequência r da Tabela através do Algoritmo de Viterbi para testar a capacidade de correção de erros do mesmo.

O diagrama de treliça mostra todas as trajetórias (caminhos) das transições de estado da máquina de estado do codificador a cada instante i de codificação, a partir do estado 00.

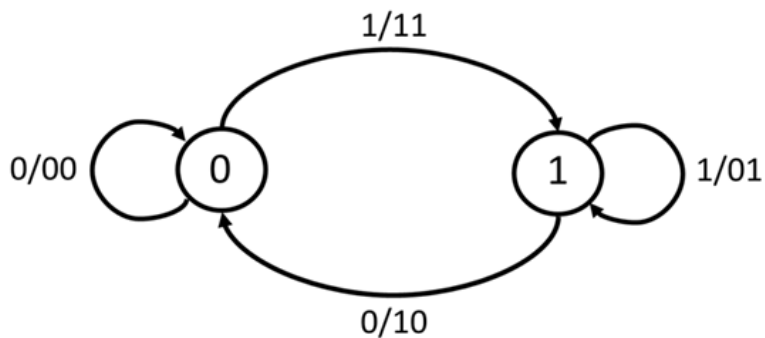


Cada ramo da treliça começa e termina em um estado, representando, assim, uma transição permitida.

Cada ramo é identificado por u/v_0v_1 , isto é, a saída v do codificador quando, ao aplicarmos u em sua entrada, a máquina de estado executa a transição representada pelo ramo em questão.

A partir do diagrama de transição de estados, inicia-se a montagem da treliça de Viterbi adequada ao codificador convolucional da Figura, para decodificação.

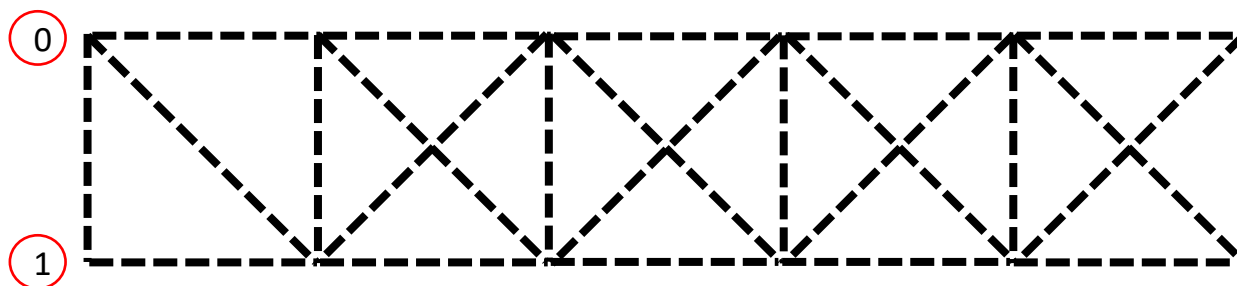
O diagrama de treliça mostra todas as trajetórias (caminhos) das transições de estado da máquina de estado do codificador a cada instante i de codificação, a partir do estado 00.



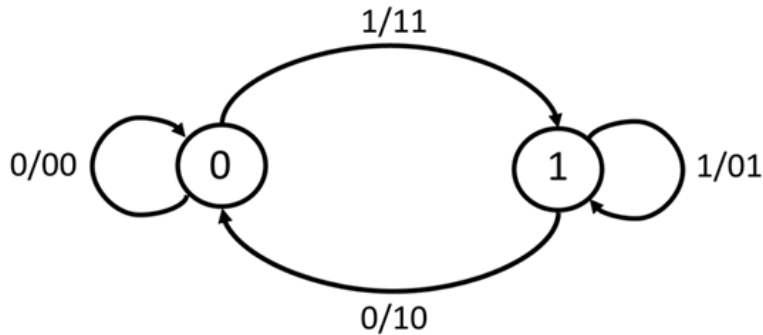
Cada ramo da treliça começa e termina em um estado, representando, assim, uma transição permitida.

Cada ramo é identificado por u/v_0v_1 , isto é, a saída v do codificador quando, ao aplicarmos u em sua entrada, a máquina de estado executa a transição representada pelo ramo em questão.

A partir do diagrama de transição de estados, inicia-se a montagem da treliça de Viterbi adequada ao codificador convolucional da Figura, para decodificação.



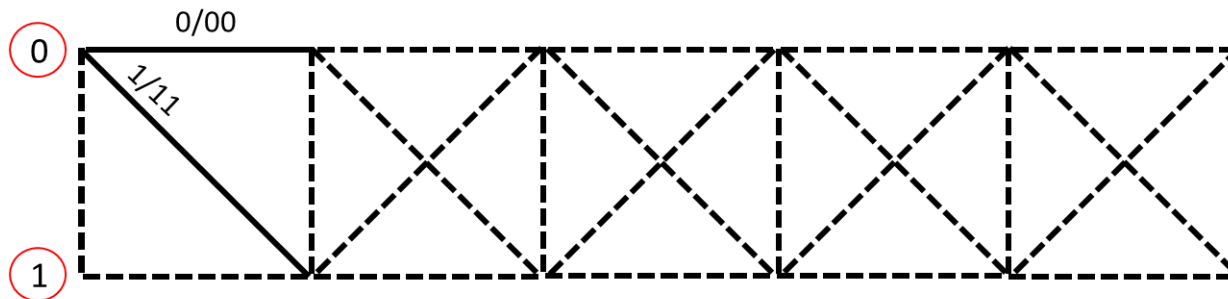
O diagrama de treliça mostra todas as trajetórias (caminhos) das transições de estado da máquina de estado do codificador a cada instante i de codificação, a partir do estado 00.



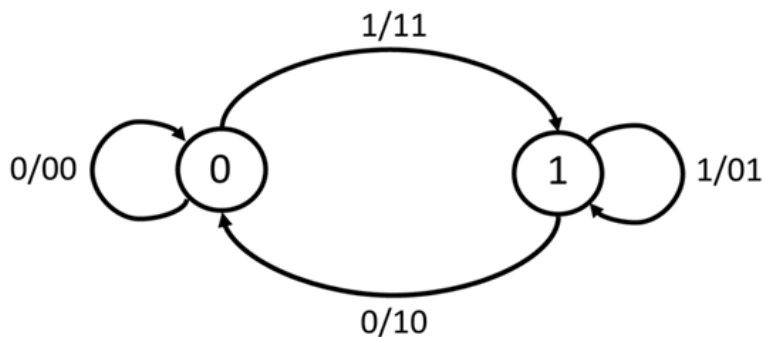
Cada ramo da treliça começa e termina em um estado, representando, assim, uma transição permitida.

Cada ramo é identificado por u/v_0v_1 , isto é, a saída v do codificador quando, ao aplicarmos u em sua entrada, a máquina de estado executa a transição representada pelo ramo em questão.

A partir do diagrama de transição de estados, inicia-se a montagem da treliça de Viterbi adequada ao codificador convolucional da Figura, para decodificação.



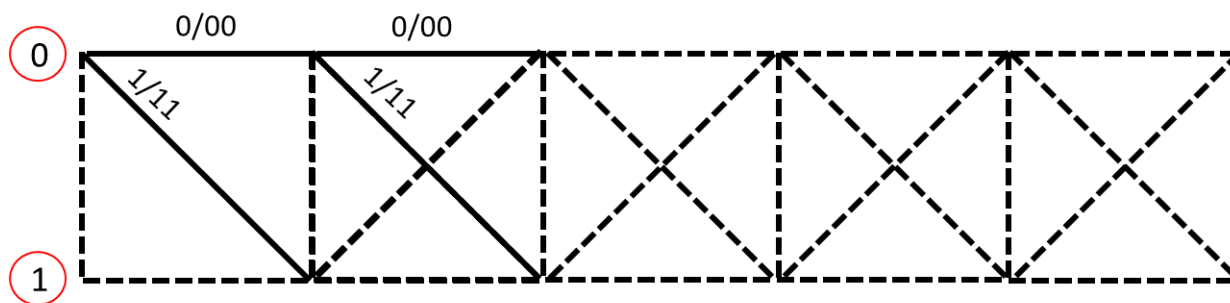
O diagrama de treliça mostra todas as trajetórias (caminhos) das transições de estado da máquina de estado do codificador a cada instante i de codificação, a partir do estado 00.



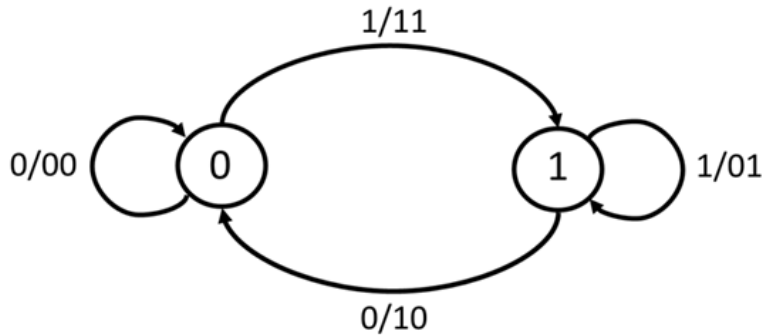
Cada ramo da treliça começa e termina em um estado, representando, assim, uma transição permitida.

Cada ramo é identificado por u/v_0v_1 , isto é, a saída v do codificador quando, ao aplicarmos u em sua entrada, a máquina de estado executa a transição representada pelo ramo em questão.

A partir do diagrama de transição de estados, inicia-se a montagem da treliça de Viterbi adequada ao codificador convolucional da Figura, para decodificação.



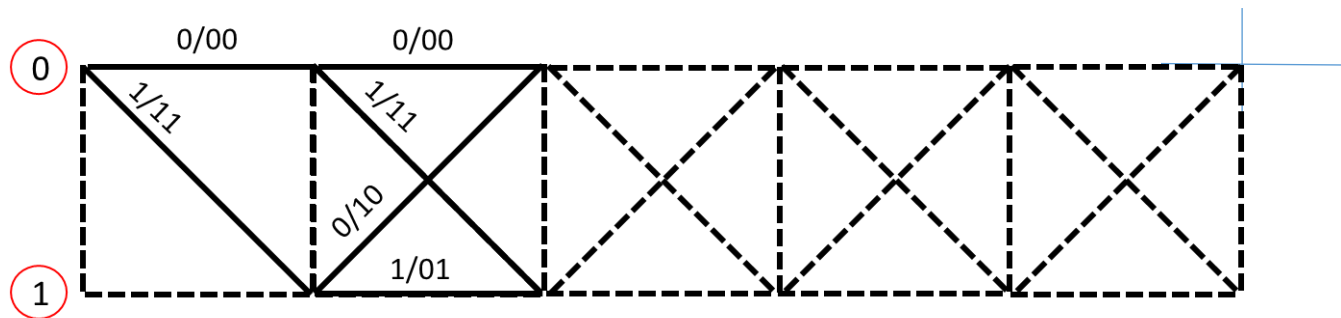
O diagrama de treliça mostra todas as trajetórias (caminhos) das transições de estado da máquina de estado do codificador a cada instante i de codificação, a partir do estado 00.



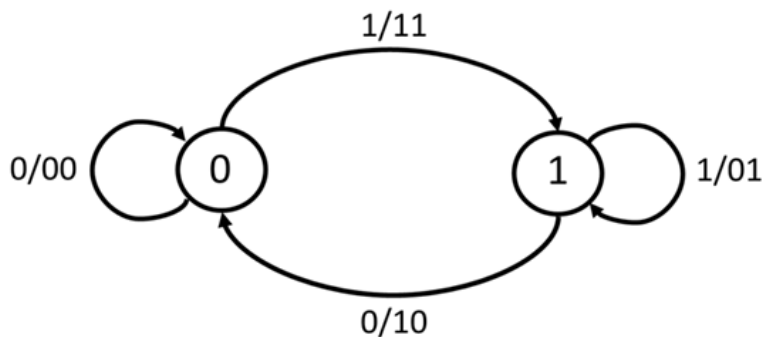
Cada ramo da treliça começa e termina em um estado, representando, assim, uma transição permitida.

Cada ramo é identificado por u/v_0v_1 , isto é, a saída v do codificador quando, ao aplicarmos u em sua entrada, a máquina de estado executa a transição representada pelo ramo em questão.

A partir do diagrama de transição de estados, inicia-se a montagem da treliça de Viterbi adequada ao codificador convolucional da Figura, para decodificação.



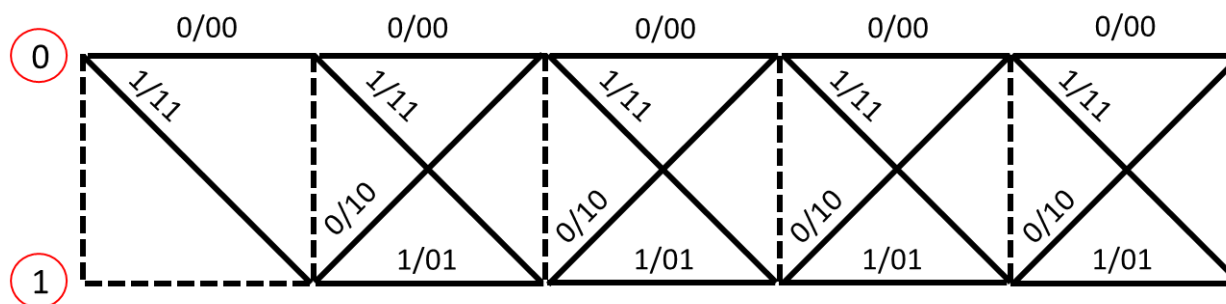
O diagrama de treliça mostra todas as trajetórias (caminhos) das transições de estado da máquina de estado do codificador a cada instante i de codificação, a partir do estado 00.



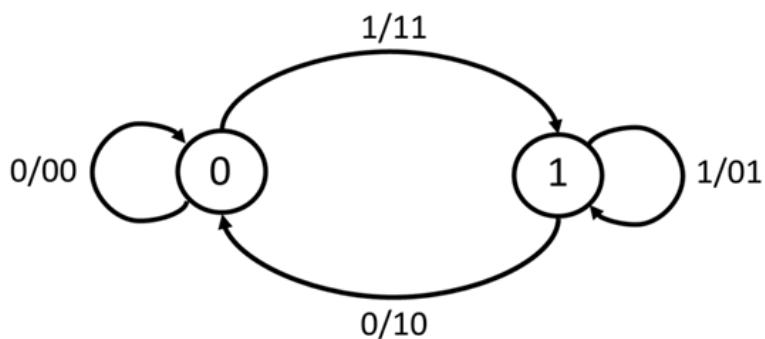
Cada ramo da treliça começa e termina em um estado, representando, assim, uma transição permitida.

Cada ramo é identificado por u/v_0v_1 , isto é, a saída v do codificador quando, ao aplicarmos u em sua entrada, a máquina de estado executa a transição representada pelo ramo em questão.

A partir do diagrama de transição de estados, inicia-se a montagem da treliça de Viterbi adequada ao codificador convolucional da Figura, para decodificação.



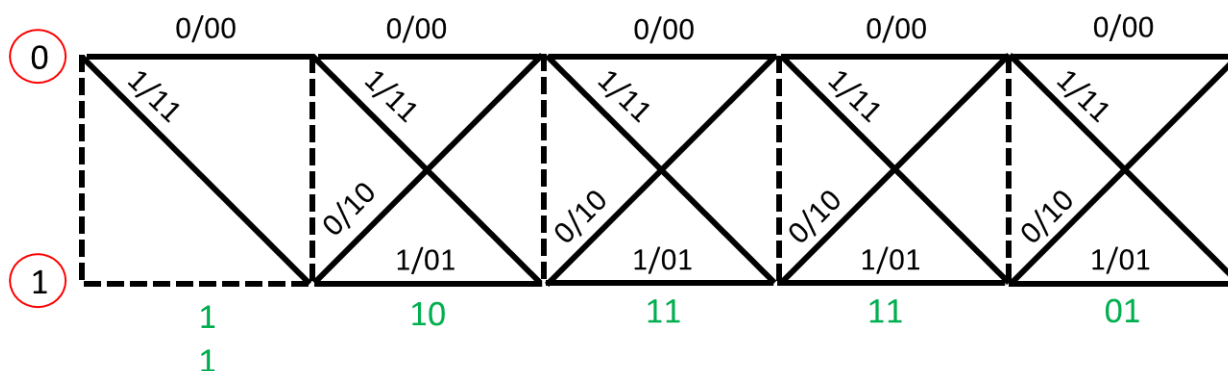
O diagrama de treliça mostra todas as trajetórias (caminhos) das transições de estado da máquina de estado do codificador a cada instante i de codificação, a partir do estado 00.



Cada ramo da treliça começa e termina em um estado, representando, assim, uma transição permitida.

Cada ramo é identificado por u/v_0v_1 , isto é, a saída v do codificador quando, ao aplicarmos u em sua entrada, a máquina de estado executa a transição representada pelo ramo em questão.

A partir do diagrama de transição de estados, inicia-se a montagem da treliça de Viterbi adequada ao codificador convolucional da Figura, para decodificação.



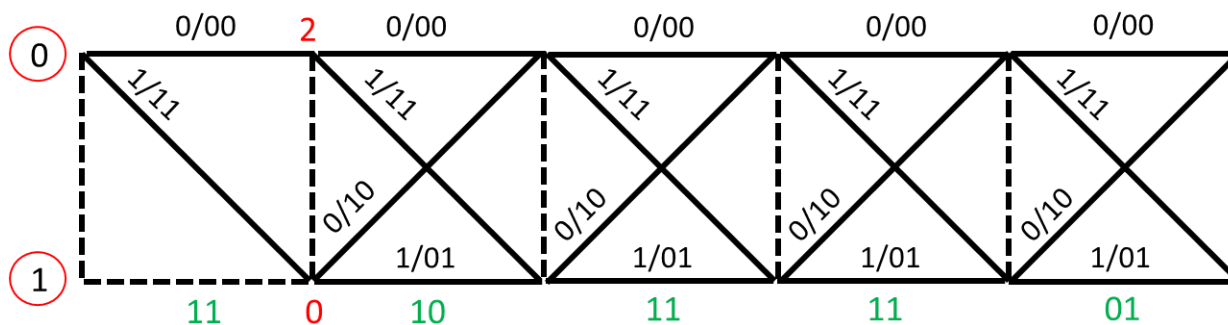
A **técnica de decodificação** consiste em acumular em cada nó da treliça as Distâncias de Hamming entre a saída v do codificador e a sequência r recebida a cada instante i .

Se mais de um caminho chega a um nó “mata-se” aqueles de maior **métrica** (maior distância acumulada) – caminhos marcados com **x** na Figura a seguir – ficando apenas aquele de menor métrica, denominado de **caminho sobrevivente**.

A métrica acumulada de cada caminho encontra-se em **negrito** à direita de cada nó, na Figura.

Métricas em **vermelho** representam ramos que incidem no nó “por cima” e métricas em **azul** representam ramos que incidem no nó “por baixo”, já que, no máximo 2 ramos incidem em um nó para este decodificador.

Métricas sublinhadas representam métricas de caminhos sobreviventes.



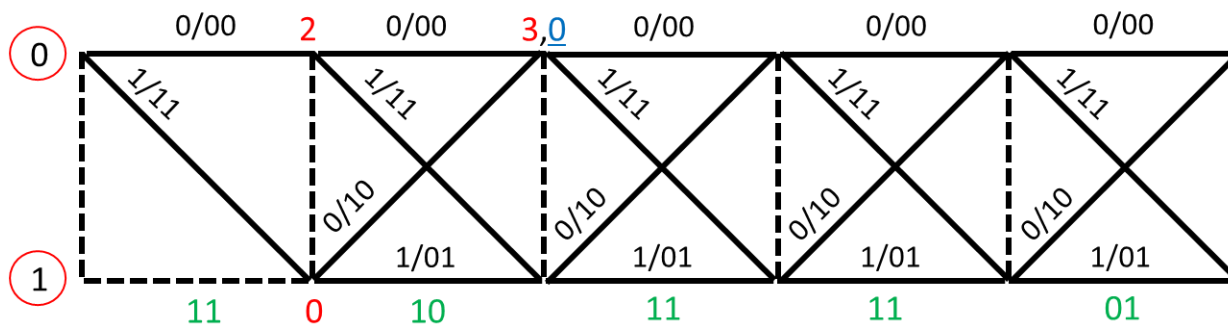
A **técnica de decodificação** consiste em acumular em cada nó da treliça as Distâncias de Hamming entre a saída v do codificador e a sequência r recebida a cada instante i .

Se mais de um caminho chega a um nó “mata-se” aqueles de maior **métrica** (maior distância acumulada) – caminhos marcados com **x** na Figura a seguir – ficando apenas aquele de menor métrica, denominado de **caminho sobrevivente**.

A métrica acumulada de cada caminho encontra-se em **negrito** à direita de cada nó, na Figura.

Métricas em **vermelho** representam ramos que incidem no nó “por cima” e métricas em **azul** representam ramos que incidem no nó “por baixo”, já que, no máximo 2 ramos incidem em um nó para este decodificador.

Métricas sublinhadas representam métricas de caminhos sobreviventes.



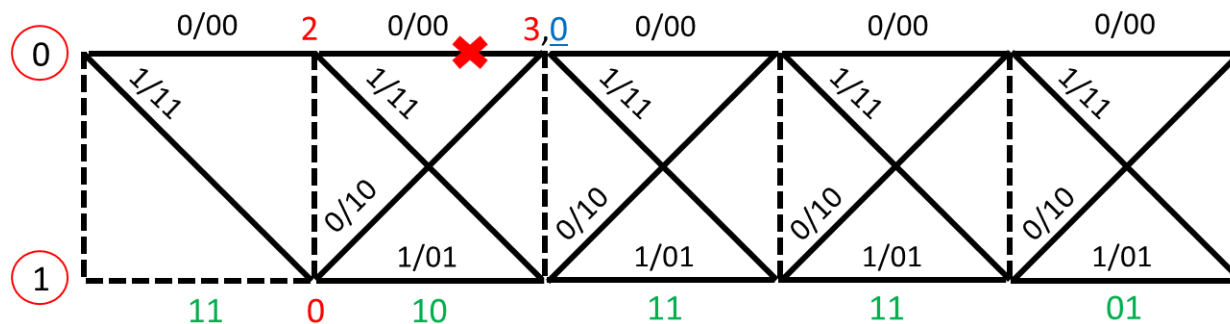
A **técnica de decodificação** consiste em acumular em cada nó da treliça as Distâncias de Hamming entre a saída v do codificador e a sequência r recebida a cada instante i .

Se mais de um caminho chega a um nó “mata-se” aqueles de maior **métrica** (maior distância acumulada) – caminhos marcados com **x** na Figura a seguir – ficando apenas aquele de menor métrica, denominado de **caminho sobrevivente**.

A métrica acumulada de cada caminho encontra-se em **negrito** à direita de cada nó, na Figura.

Métricas em **vermelho** representam ramos que incidem no nó “por cima” e métricas em **azul** representam ramos que incidem no nó “por baixo”, já que, no máximo 2 ramos incidem em um nó para este decodificador.

Métricas sublinhadas representam métricas de caminhos sobreviventes.



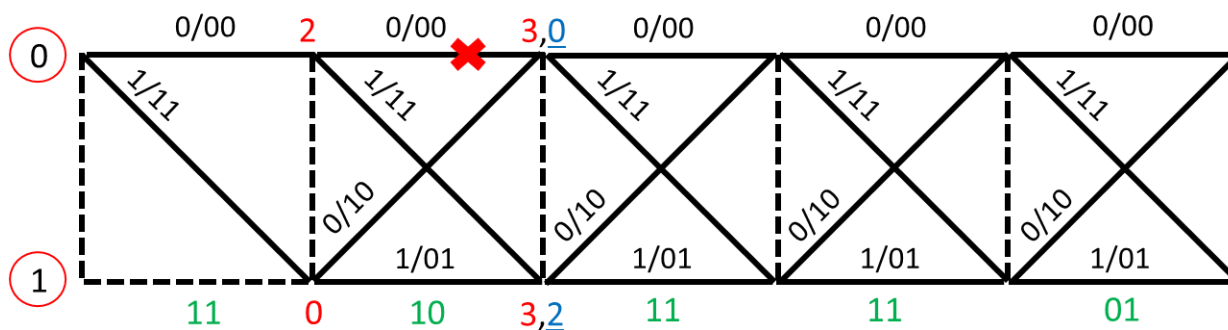
A **técnica de decodificação** consiste em acumular em cada nó da treliça as Distâncias de Hamming entre a saída v do codificador e a sequência r recebida a cada instante i .

Se mais de um caminho chega a um nó “mata-se” aqueles de maior **métrica** (maior distância acumulada) – caminhos marcados com **x** na Figura a seguir – ficando apenas aquele de menor métrica, denominado de **caminho sobrevivente**.

A métrica acumulada de cada caminho encontra-se em **negrito** à direita de cada nó, na Figura.

Métricas em **vermelho** representam ramos que incidem no nó “por cima” e métricas em **azul** representam ramos que incidem no nó “por baixo”, já que, no máximo 2 ramos incidem em um nó para este decodificador.

Métricas sublinhadas representam métricas de caminhos sobreviventes.



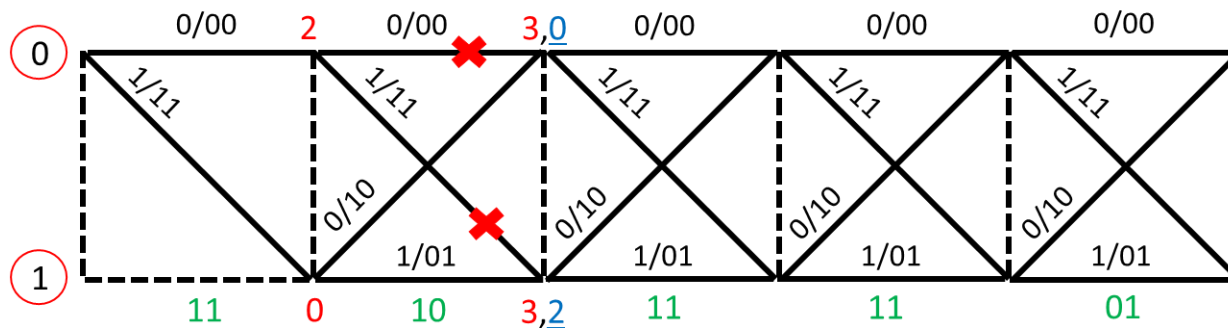
A **técnica de decodificação** consiste em acumular em cada nó da treliça as Distâncias de Hamming entre a saída v do codificador e a sequência r recebida a cada instante i .

Se mais de um caminho chega a um nó “mata-se” aqueles de maior **métrica** (maior distância acumulada) – caminhos marcados com **x** na Figura a seguir – ficando apenas aquele de menor métrica, denominado de **caminho sobrevivente**.

A métrica acumulada de cada caminho encontra-se em **negrito** à direita de cada nó, na Figura.

Métricas em **vermelho** representam ramos que incidem no nó “por cima” e métricas em **azul** representam ramos que incidem no nó “por baixo”, já que, no máximo 2 ramos incidem em um nó para este decodificador.

Métricas sublinhadas representam métricas de caminhos sobreviventes.



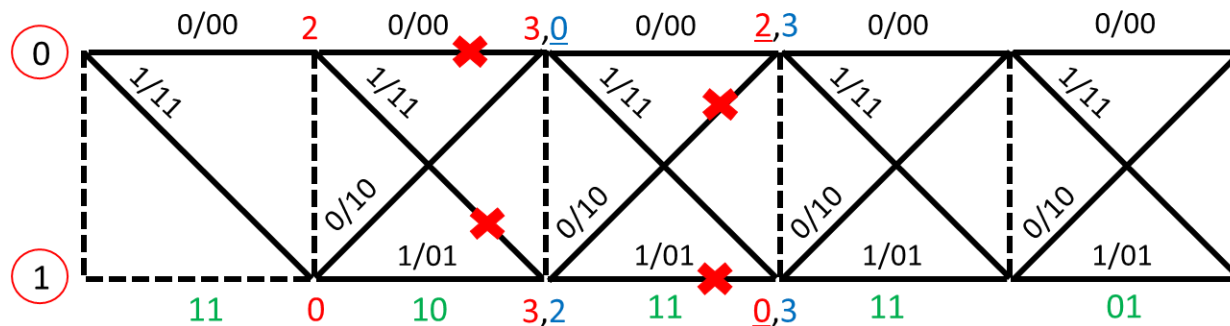
A **técnica de decodificação** consiste em acumular em cada nó da treliça as Distâncias de Hamming entre a saída v do codificador e a sequência r recebida a cada instante i .

Se mais de um caminho chega a um nó “mata-se” aqueles de maior **métrica** (maior distância acumulada) – caminhos marcados com **x** na Figura a seguir – ficando apenas aquele de menor métrica, denominado de **caminho sobrevivente**.

A métrica acumulada de cada caminho encontra-se em **negrito** à direita de cada nó, na Figura.

Métricas em **vermelho** representam ramos que incidem no nó “por cima” e métricas em **azul** representam ramos que incidem no nó “por baixo”, já que, no máximo 2 ramos incidem em um nó para este decodificador.

Métricas sublinhadas representam métricas de caminhos sobreviventes.



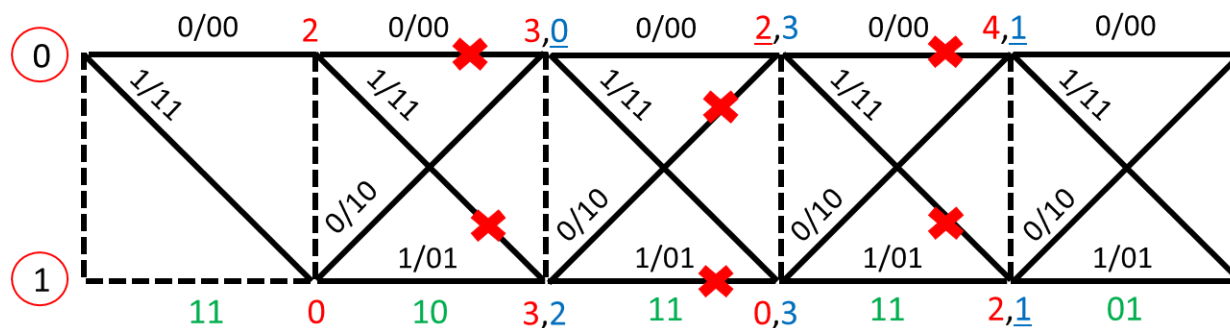
A **técnica de decodificação** consiste em acumular em cada nó da treliça as Distâncias de Hamming entre a saída v do codificador e a sequência r recebida a cada instante i .

Se mais de um caminho chega a um nó “mata-se” aqueles de maior **métrica** (maior distância acumulada) – caminhos marcados com **x** na Figura a seguir – ficando apenas aquele de menor métrica, denominado de **caminho sobrevivente**.

A métrica acumulada de cada caminho encontra-se em **negrito** à direita de cada nó, na Figura.

Métricas em **vermelho** representam ramos que incidem no nó “por cima” e métricas em **azul** representam ramos que incidem no nó “por baixo”, já que, no máximo 2 ramos incidem em um nó para este decodificador.

Métricas sublinhadas representam métricas de caminhos sobreviventes.



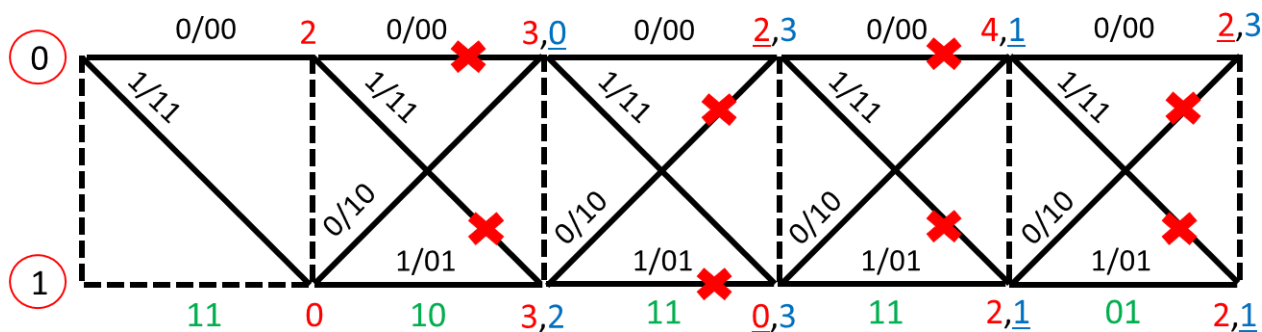
A **técnica de decodificação** consiste em acumular em cada nó da treliça as Distâncias de Hamming entre a saída v do codificador e a sequência r recebida a cada instante i .

Se mais de um caminho chega a um nó “mata-se” aqueles de maior **métrica** (maior distância acumulada) – caminhos marcados com **x** na Figura a seguir – ficando apenas aquele de menor métrica, denominado de **caminho sobrevivente**.

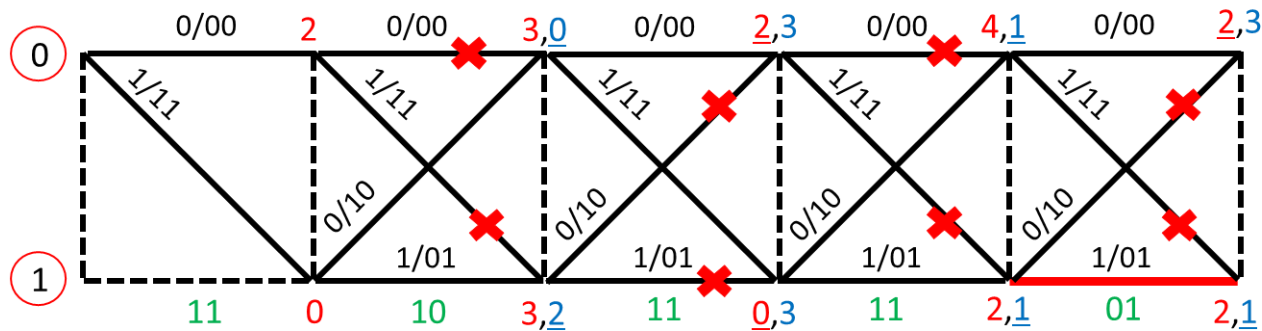
A métrica acumulada de cada caminho encontra-se em **negrito** à direita de cada nó, na Figura.

Métricas em **vermelho** representam ramos que incidem no nó “por cima” e métricas em **azul** representam ramos que incidem no nó “por baixo”, já que, no máximo 2 ramos incidem em um nó para este decodificador.

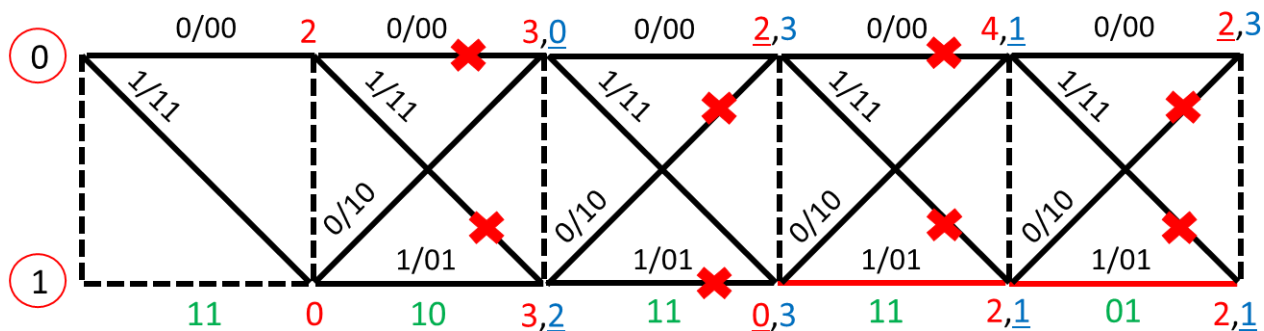
Métricas sublinhadas representam métricas de caminhos sobreviventes.



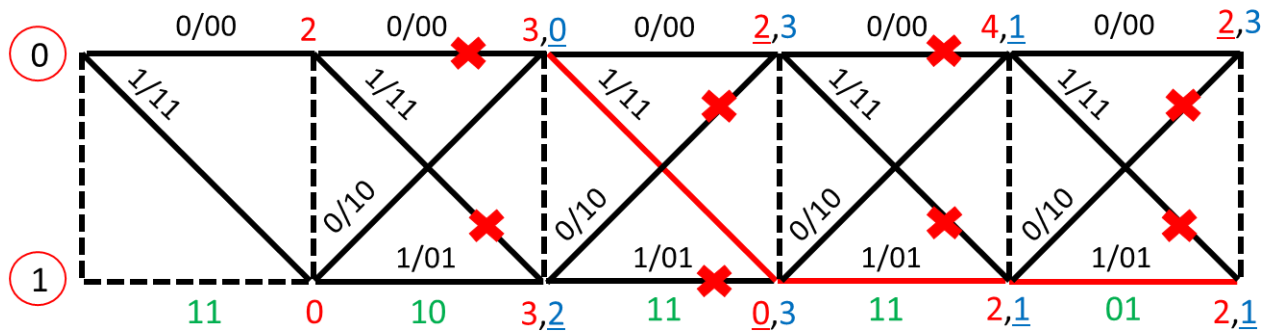
A **decodificação final** é iniciada a partir do caminho sobrevivente de menor métrica acumulada, identificando cada ramo sobrevivente da direita para a esquerda na treliça.



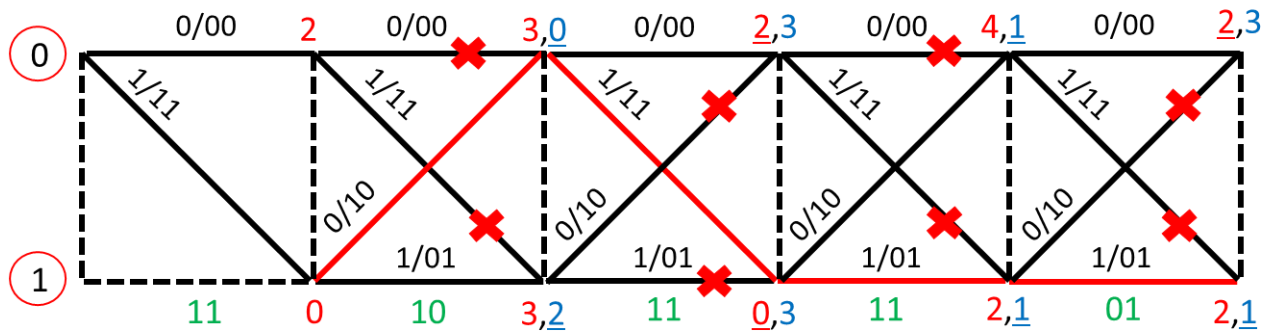
A **decodificação final** é iniciada a partir do caminho sobrevivente de menor métrica acumulada, identificando cada ramo sobrevivente da direita para a esquerda na treliça.



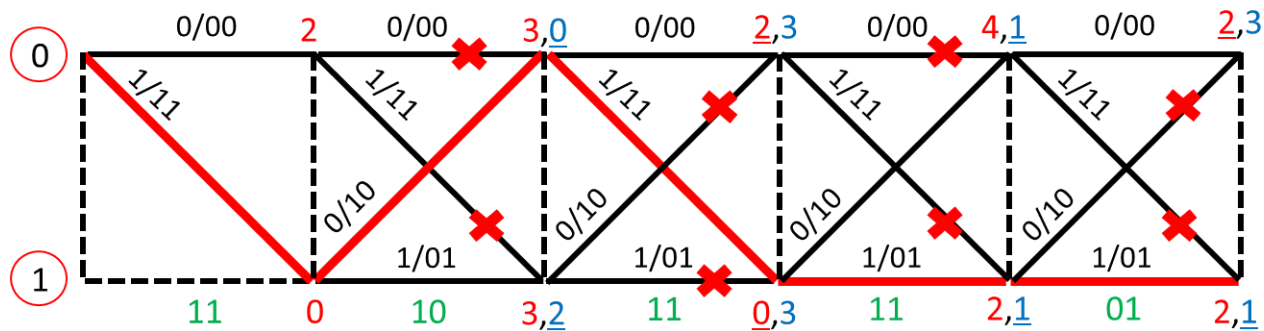
A **decodificação final** é iniciada a partir do caminho sobrevivente de menor métrica acumulada, identificando cada ramo sobrevivente da direita para a esquerda na treliça.



A **decodificação final** é iniciada a partir do caminho sobrevivente de menor métrica acumulada, identificando cada ramo sobrevivente da direita para a esquerda na treliça.



A **decodificação final** é iniciada a partir do caminho sobrevivente de menor métrica acumulada, identificando cada ramo sobrevivente da direita para a esquerda na treliça.



A **decodificação final** é iniciada a partir do caminho sobrevivente de menor métrica acumulada, identificando cada ramo sobrevivente da direita para a esquerda na treliça.

Ao lermos o valor de u nos identificadores u/v_0v_1 de cada ramo sobrevivente na Figura, verificamos que a sequência originalmente transmitida foi $u = [01100]$, o que concorda com u originalmente transmitido.

Portanto, o **decodificador identificou e corrigiu** o erro ocorrido.

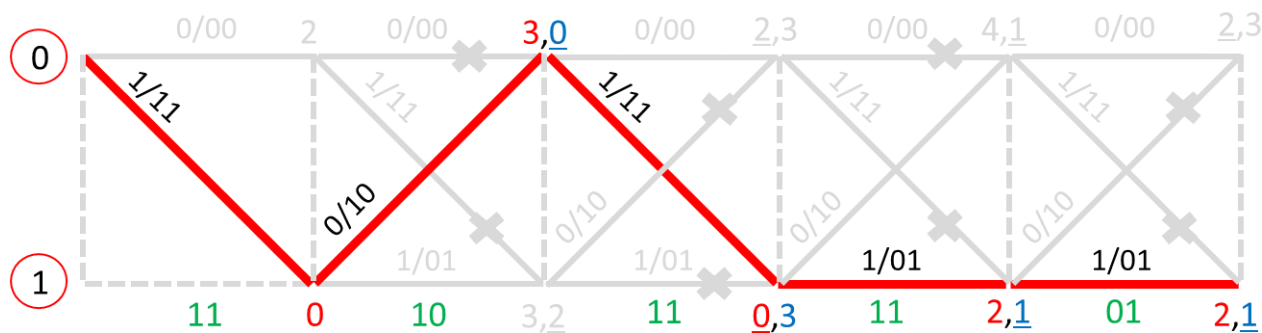
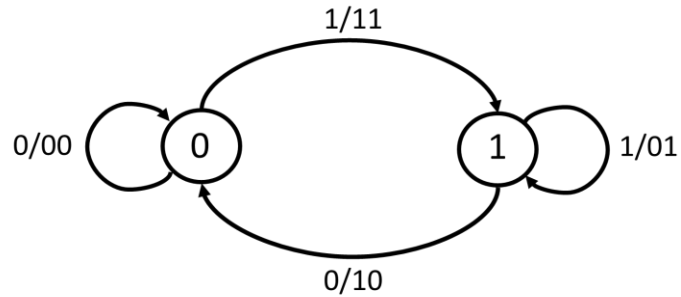


Diagrama de Transição de Estados

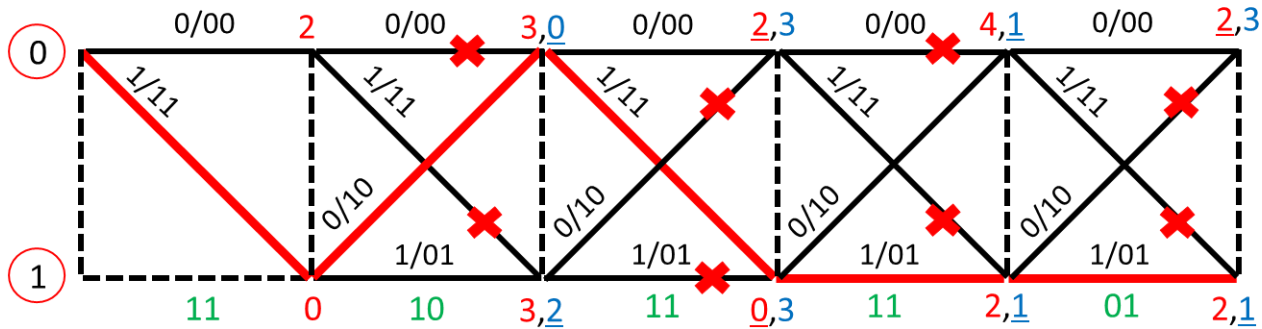


Exemplo de codificação para o codificador da Figura

u	1	0	1	1	1
D _{atual}	0	1	0	1	1
D _{futuro}	1	0	1	1	1
v	11	10	11	01	01
r	11	10	11	11	01

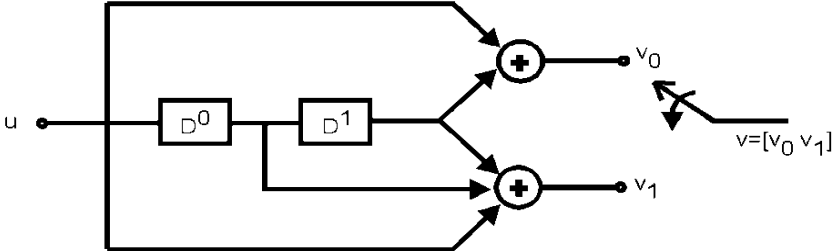


Diagrama de Treliza do Decodificador de Viterbi



Exemplo 2:

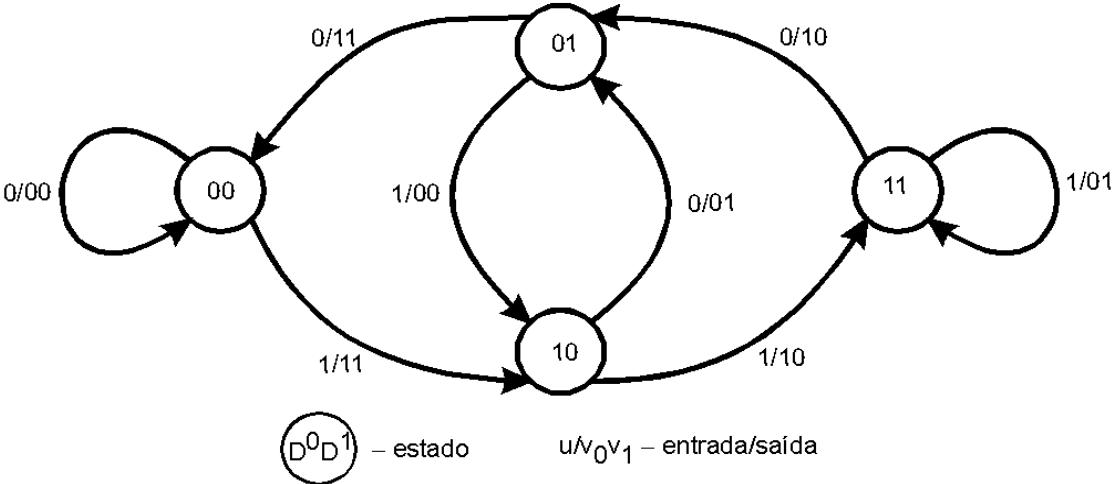
A Figura abaixo apresenta o diagrama de transição de estados do codificador convolucional da Figura ao lado. Na Figura, cada círculo representa um estado D^0D^1 dentre os $2^K = 4$ possíveis estados.



O diagrama é construído a partir dos estados individuais considerando as transições permitidas a partir de cada estado como consequência do valor lógico de u .

Por exemplo, suponhamos que a máquina de estado encontre-se no estado **10** (i.e., $D^0 = 1$ e $D^1 = 0$ na Figura acima).

- ⇒ Se $u = 1$ a saída resultante é $v = 10$ e, após a borda de descida do *clock*, a máquina vai para o estado **11**.
- ⇒ Se $u = 0$ a saída resultante é $v = 01$ e, após a borda de descida do *clock*, a máquina vai para o estado **01**.



Dada uma seqüência u a ser codificada, a saída v no codificador de um transmissor digital é enviada ao receptor através do canal de transmissão, sendo recebida como uma seqüência r .

Se nenhuma degradação de sinal ocorreu no canal de transmissão, $r = v$.

A Tabela a seguir mostra uma possível seqüência u e a resultante seqüência v para o codificador do Exemplo 2.

É mostrada também a trajetória do estado D^0D^1 , à medida que u é codificada, partindo inicialmente do estado 00.

Assumindo que v seja enviado através de um canal de transmissão com ruído/interferência, a Tabela apresenta uma possível seqüência r recebida com 2 erros.

Codificação					
$u =$	0	1	1	0	0
D^0D^1 atual =	00	00	10	11	01
D^0D^1 futuro =	00	10	11	01	00
$v =$	00	11	10	10	11
$r =$	0 I	11	1 I	10	11

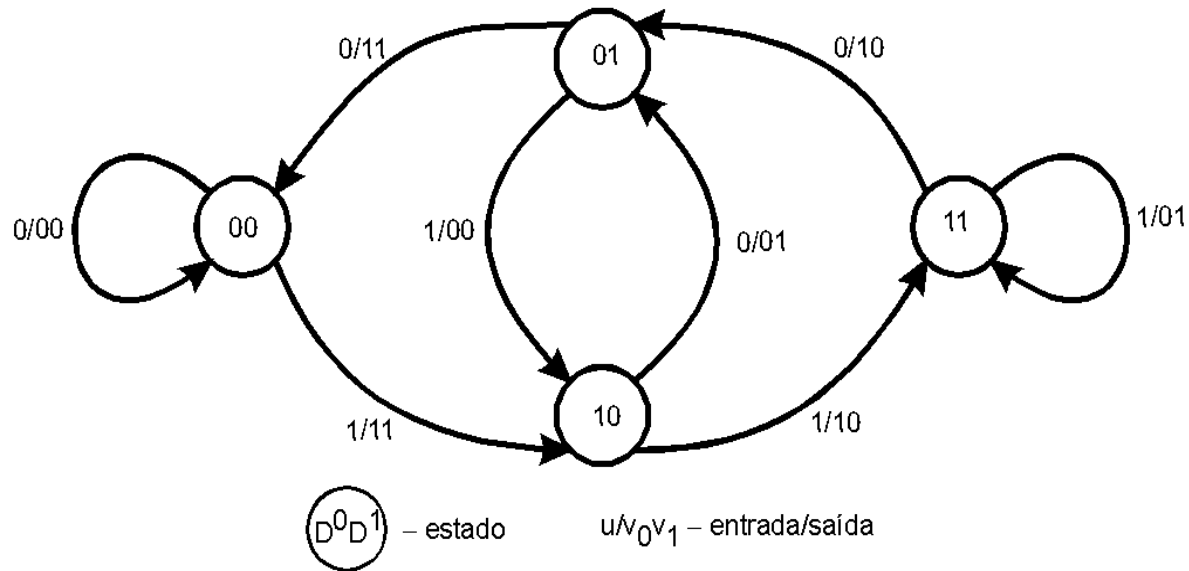


Diagrama de transição de estados do codificador convolucional.

No receptor digital, o decodificador utiliza um algoritmo de decodificação baseado no princípio de mínima distância (MLSE – *maximum likelihood sequence detector*) denominado [Algoritmo de Viterbi](#). (Vamos decodificar a sequência r da Tabela através do Algoritmo de Viterbi para testar a capacidade de correção de erros do mesmo.)

A Figura a seguir apresenta o [diagrama de treliça](#) utilizado pelo Decodificador de Viterbi adequado a este codificador convolucional.

⇒ O diagrama de treliça mostra todas as trajetórias (caminhos) das transições de estado da máquina de estado do codificador a cada instante i de codificação, a partir do estado 00.

⇒ Cada ramo da treliça começa e termina em um estado, representando, assim, uma transição permitida.

⇒ Cada ramo é identificado por u/v_0v_1 , isto é, a saída v do codificador quando, ao aplicarmos u em sua entrada, a máquina de estado executa a transição representada pelo ramo em questão.

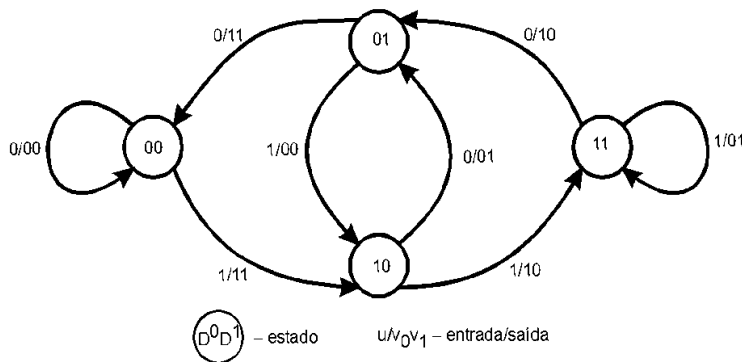


Diagrama de transição de estados do codificador convolucional.

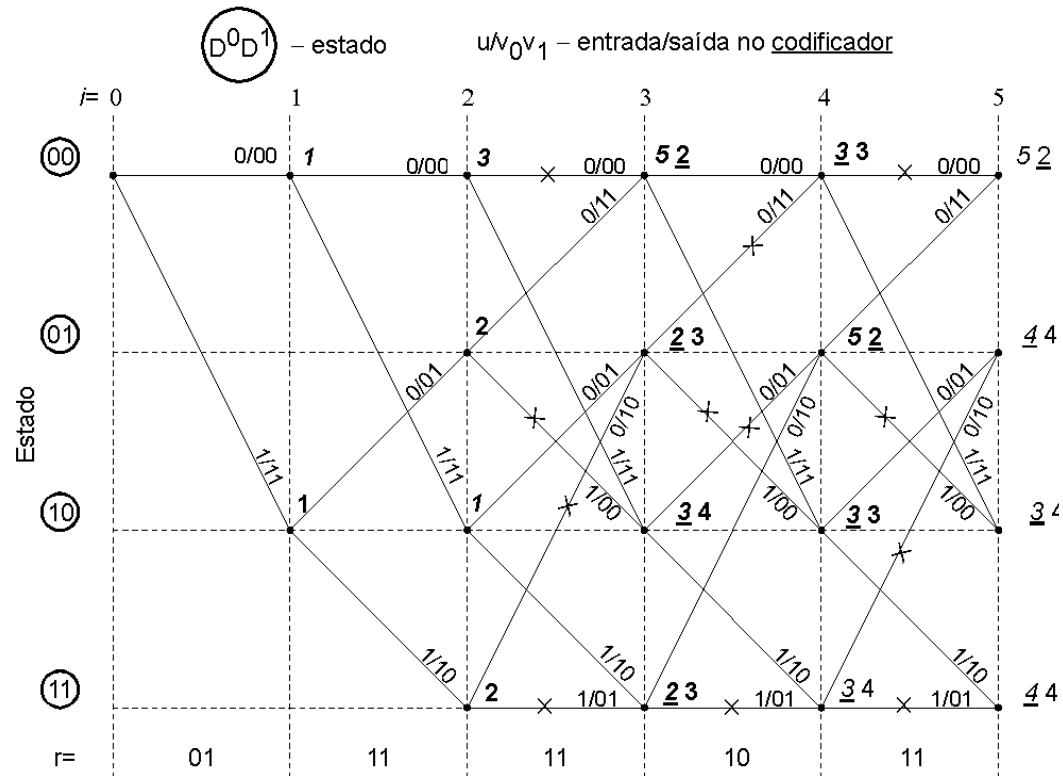


Diagrama de Treliça do Decodificador de Viterbi para o codificador convolucional.

A **técnica de decodificação** consiste em acumular em cada nó da treliça as Distâncias de Hamming entre a saída v do codificador e a sequência r recebida a cada instante i .

Se mais de um caminho chega a um nó “mata-se” aqueles de maior **métrica** (maior distância acumulada) – caminhos marcados com **x** na Figura – ficando apenas aquele de menor métrica, denominado de **caminho sobrevivente**.

A métrica acumulada de cada caminho encontra-se em **negrito** à direita de cada nó na Figura.

Métricas sublinhadas representam métricas de caminhos sobreviventes.

Métricas em *itálico* representam ramos que incidem no nó “por cima” e métricas em não-*itálico* representam ramos que incidem no nó “por baixo”, já que, no máximo 2 ramos incidem em um nó para este decodificador.

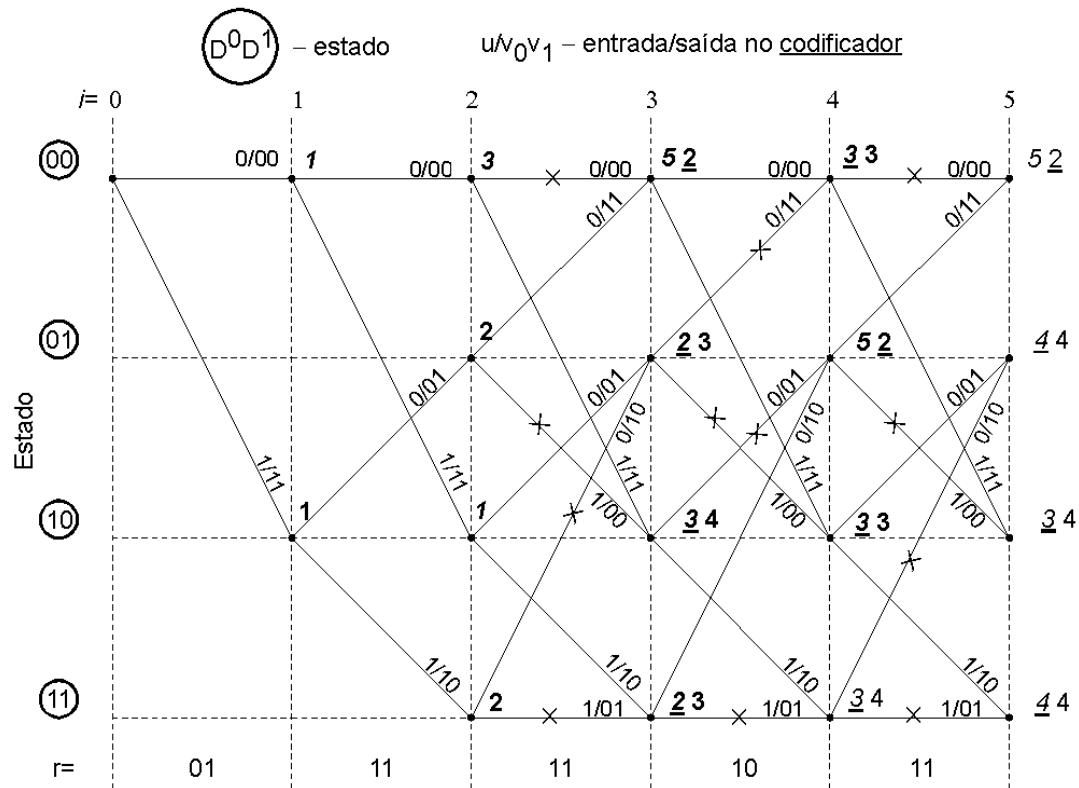


Diagrama de Treliça do Decodificador de Viterbi para o codificador convolucional.

A decodificação final é iniciada a partir do caminho sobrevivente de menor métrica acumulada, identificando cada ramo sobrevivente da direita para a esquerda na treliça, conforma mostra a Figura.

Ao lermos o valor de u nos identificadores u/v_0v_1 de cada ramo sobrevivente na Figura, verificamos que a sequência originalmente transmitida foi $u = [01100]$, o que concorda com u mostrado na Tabela (efetivamente transmitido).

Portanto, o decodificador identificou e corrigiu os 2 erros.

