of obtaining the transfer function and other related performance measures are described below.

2.7.1 Transfer Function of Convolutional Code

The analysis of convolutional codes is generally difficult to perform because traditional algebraic and combinatorial techniques cannot be applied. These heuristically constructed codes can be analyzed through their transfer functions. By utilizing the state diagram, the transfer function can be obtained. With the transfer function, code properties such as distance properties and the error rate performance can be easily calculated. To obtain the transfer function, the following rules are applied:

1. Break the all-zero (initial) state of the state diagram into a start state and an end state. This will be called the modified state diagram.
2. For every branch of the modified state diagram, assign the symbol D with its exponent equal to the Hamming weight of the output bits.
3. For every branch of the modified state diagram, assign the symbol J.
4. Assign the symbol N to the branch of the modified state diagram, if the branch transition is caused by an input bit 1.

For the state diagram in Figure 2.4, the modified state diagram is shown in Figure 2.14.

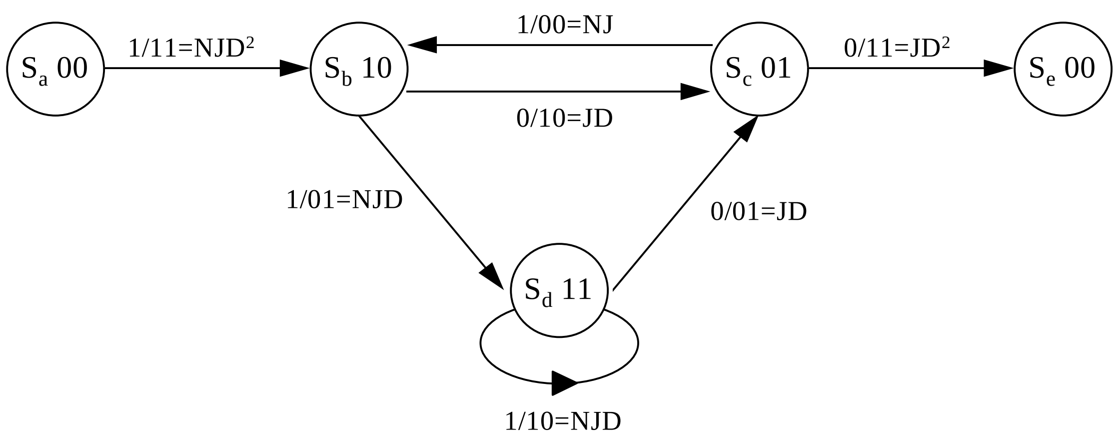

Figure 2.14: The modified state diagram of Figure 2.4 where S_a is the start state and S_e is the end state.

Nodal equations are obtained for all the states except for the start state in Figure 2.14. These results are

$$S_b = NJD^2 S_a + NJS_c$$
$$S_c = JDS_b + JDS_d$$
$$S_d = NJDS_b + NJDS_d$$
$$S_e = JD^2 S_c$$

The transfer function is defined to be

$$T(D, N, J) = \frac{S_{end}(D, N, J)}{S_{start}(D, N, J)} \quad (2.29)$$

and for Figure 2.14,

$$T(D, N, J) = \frac{S_e}{S_a}$$

By substituting and rearranging,

$$T(D, N, J) = \frac{NJ^3 D^5}{1 - (NJ + NJ^2)D} \quad \text{(closed form)}$$

$$= NJ^3 D^5 + (N^2 J^4 + N^2 J^5) D^6 + (N^3 J^5 + 2N^3 J^6 + N^3 J^7) D^7 + \cdots$$

(expanded polynomial form)

2.7.1.1 Distance Properties

The free distance between a pair of convolutional codewords is the Hamming distance between the pair of codewords. The minimum free distance, d_{free}, is the minimum Hamming distance between all pairs of complete convolutional codewords and is defined as

$$d_{free} = \min\{d(\mathbf{y_1}, \mathbf{y_2}) | \mathbf{y_1} \neq \mathbf{y_2}\} \quad \text{[Wic95]} \quad (2.30)$$

$$= \min\{w(\mathbf{y}) | \mathbf{y} \neq \mathbf{0}\} \quad \text{[Wic95]} \quad (2.31)$$

where d(•,•) is the Hamming distance between a pair of convolutional codewords and w(•) is the Hamming distance between a convolutional codeword and the all-zero codeword (the weight of the codeword). The minimum free distance corresponds to the ability of the convolutional code to estimate the best decoded bit sequence. As d_{free} increases, the performance of the convolutional code also increases. This characteristic is similar to the minimum distance for block codes. From the transfer function, the minimum free distance is identified as the lowest exponent of D. From the above transfer function for Figure 2.14, $d_{free} = 5$. Also, if N and J are set to 1, the coefficients of D^i's represent the number of paths through the trellis with weight D^i. More information about the codeword is obtained from observing the exponents of N and J. For a codeword, the exponent of N indicates the number of 1s in the input sequence, and the exponent of J indicates the length of the path that merges with the all-zero path for the first time [Pro95].

2.7.1.2 Error Probabilities

There are two error probabilities associated with convolutional codes, namely first event and bit error probabilities. The first event error probability, P_e, is the probability that an error begins at a particular time. The bit error probability, P_b, is the average