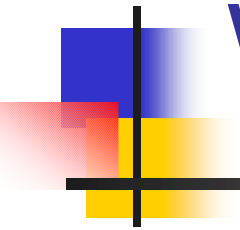


What is this “Viterbi Decoding”

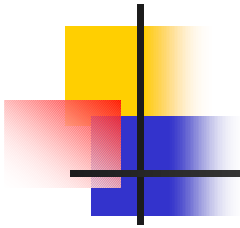
Andrew A. Lamb

MIT

Laboratory for Computer Science

Computer Architecture Group

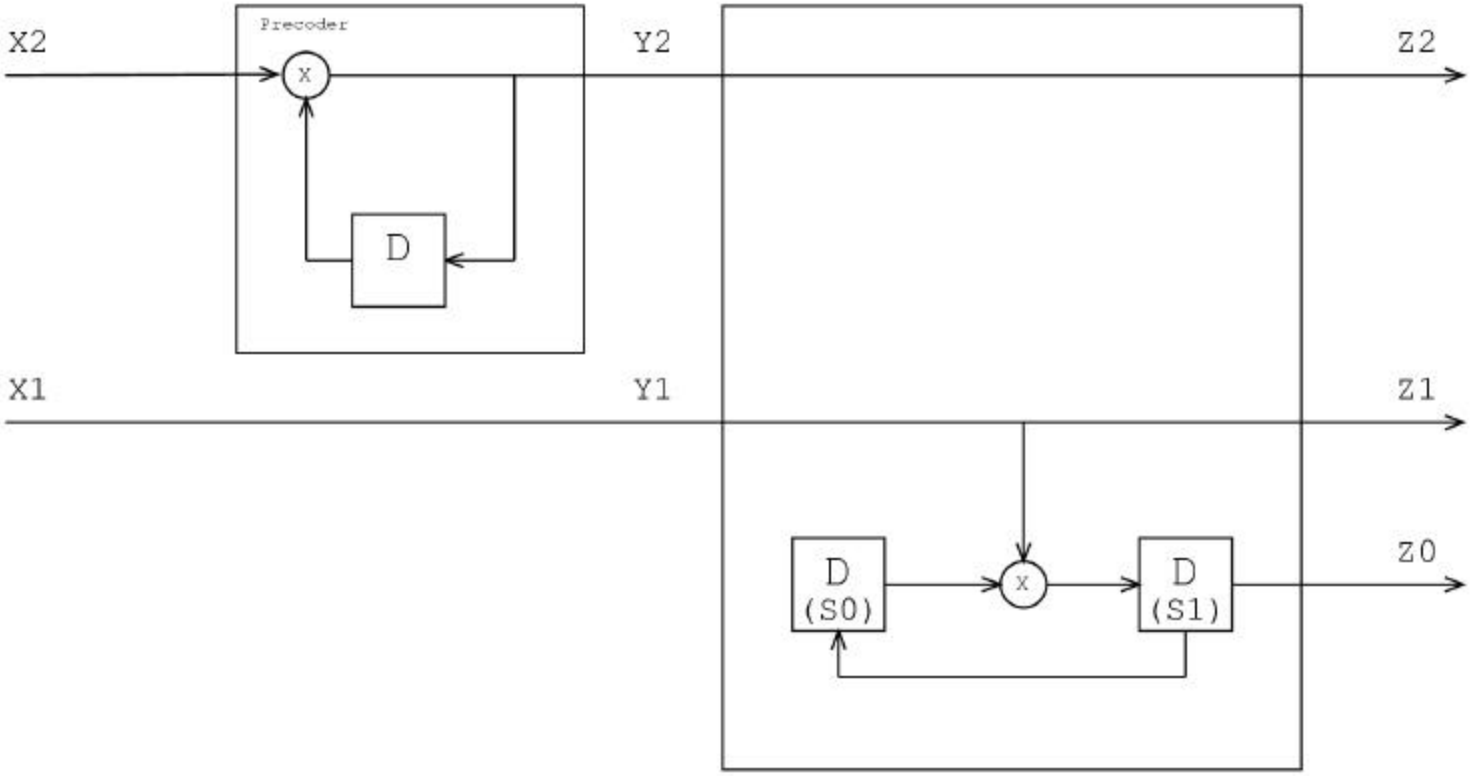
(8/2/2002)



State Machine/Trellis Encoding?

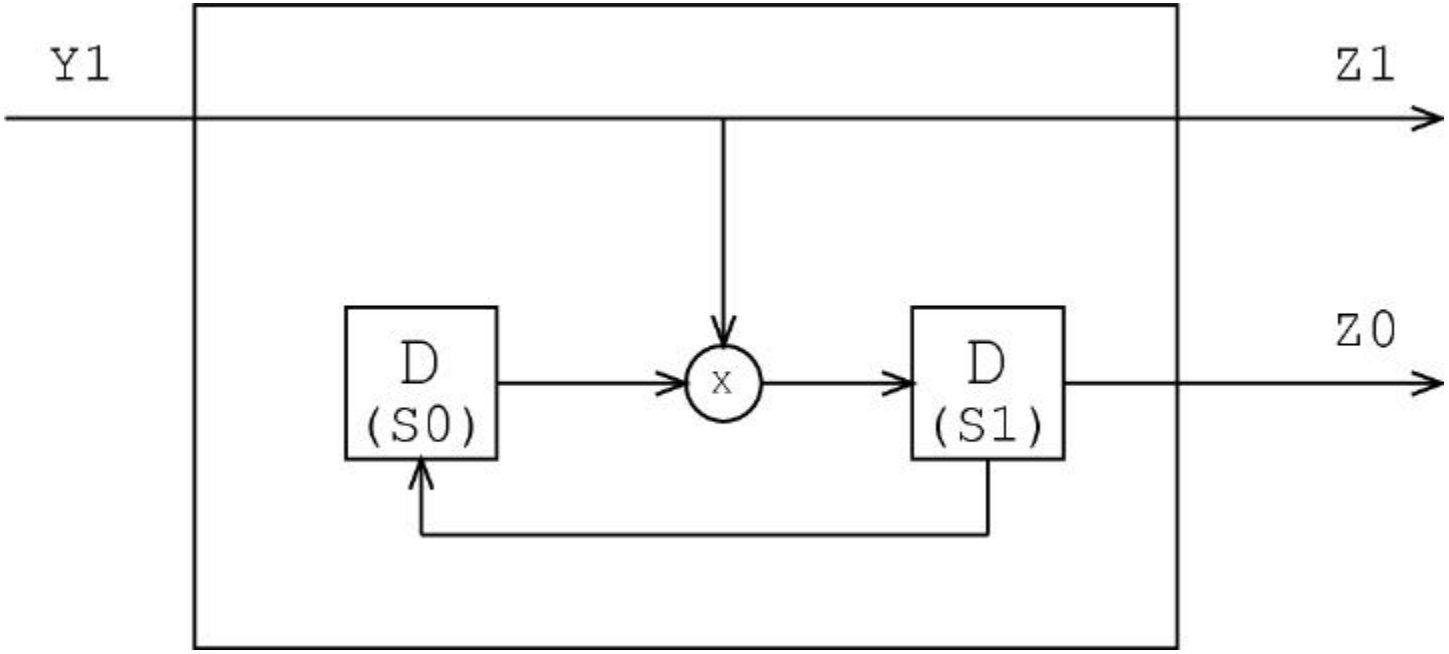
- You can also represent the convolution encoder as a finite state machine
- The possible state evolution of the convolution encoder traces out a “trellis” like structure in state-time space.

Convolutional Encoder

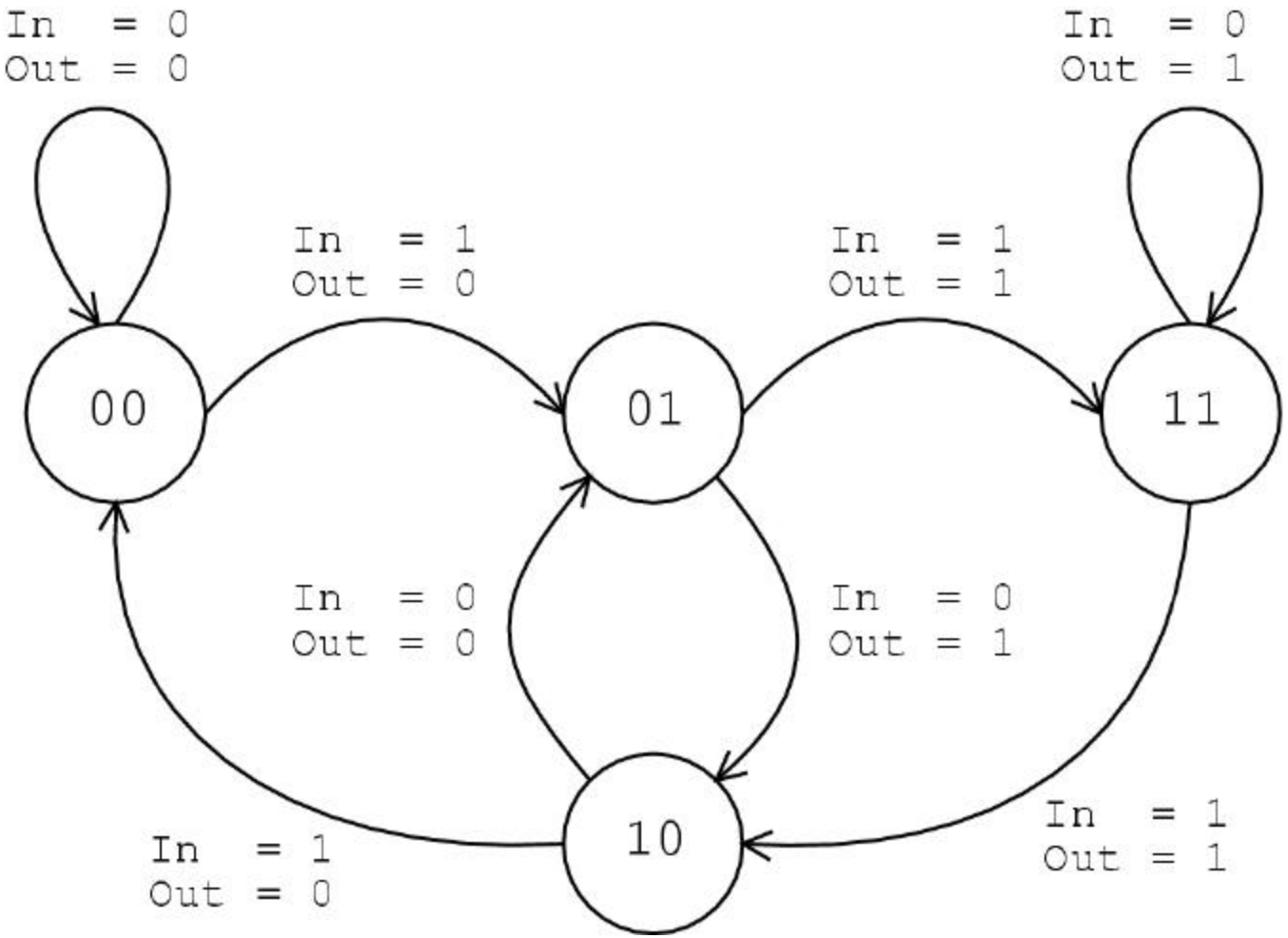


Source: ASTC Standard A53, revision B

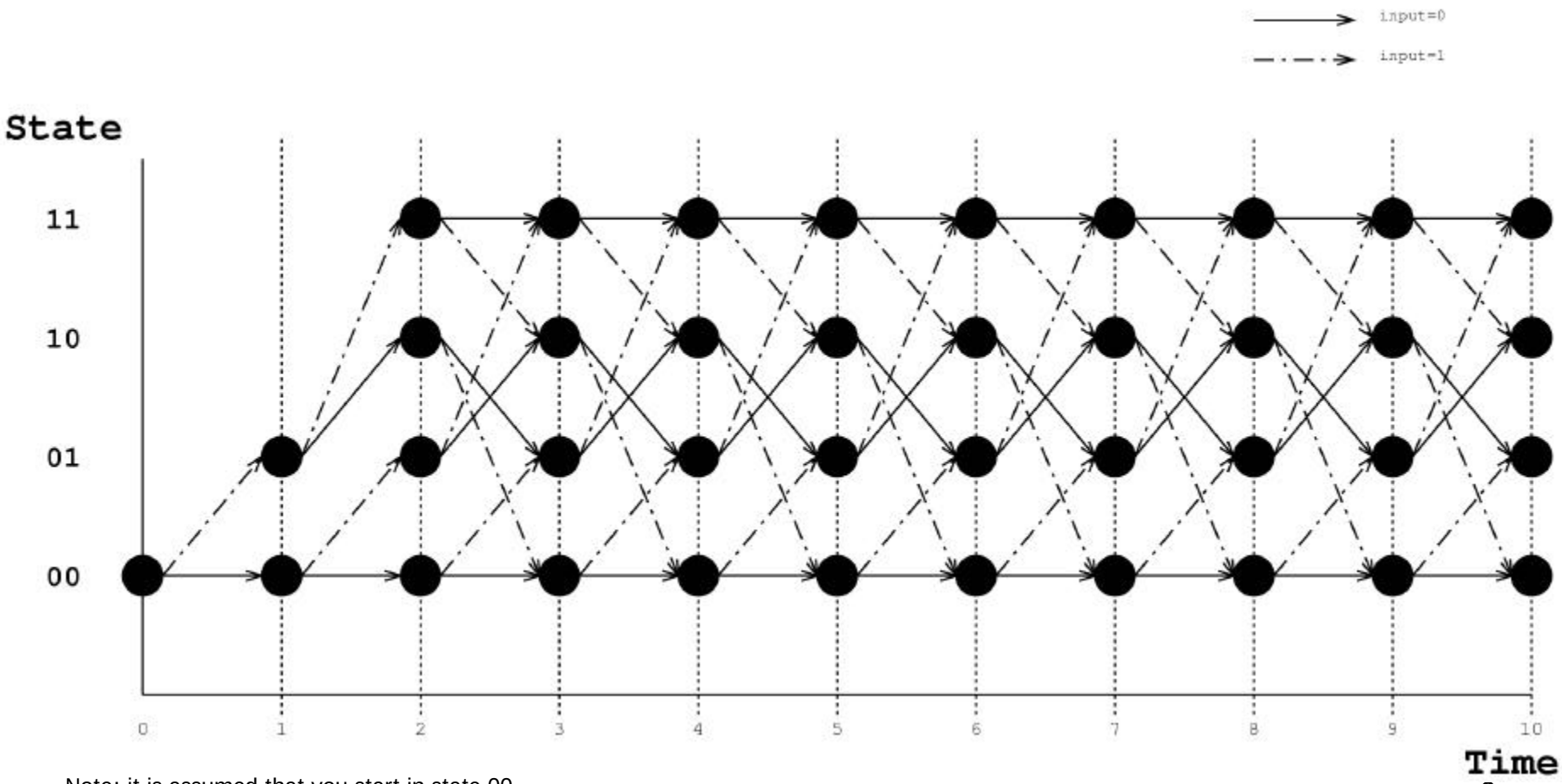
Interesting Part



State Transition Diagram



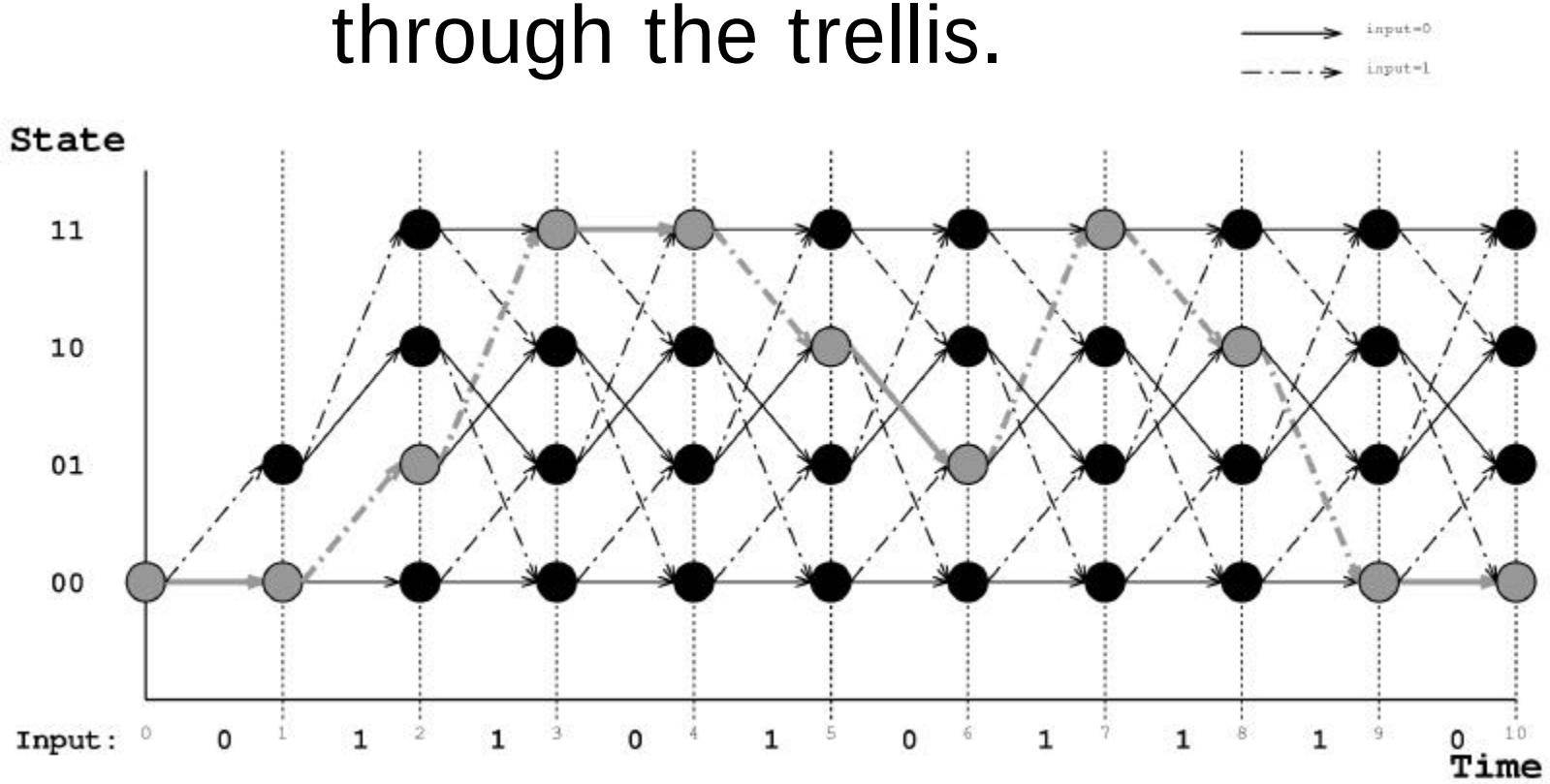
Trellis Diagram



Note: it is assumed that you start in state 00

Trellis Paths

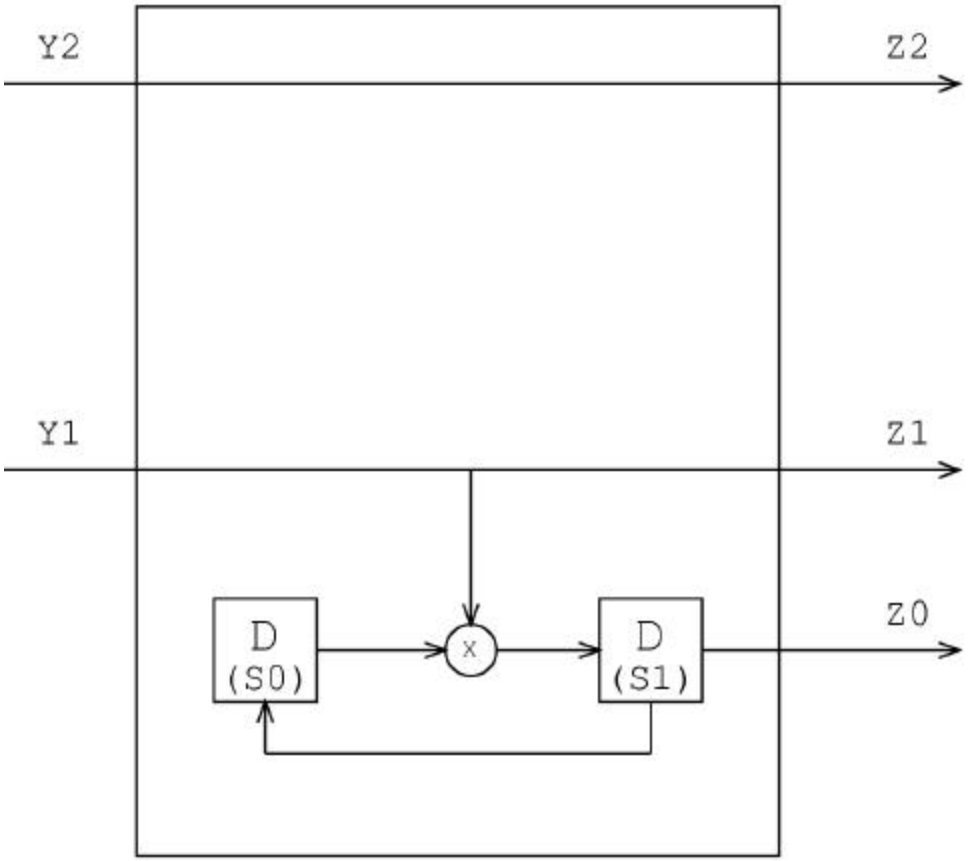
- Each set of inputs traces a unique path through the trellis.



Input:

01101
01110

Symbol Mapping

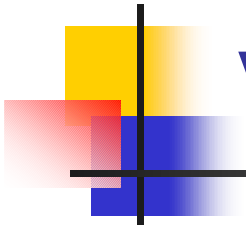


MAP

Z2	Z1	Z0	R
0	0	0	−7
0	0	1	−5
0	1	0	−3
0	1	1	−1
1	0	0	+1
1	0	1	+3
1	1	0	+5
1	1	1	+7

Trellis Decoder

- Use Viterbi Algorithm to undo Convolution Coding (not signal → symbol unmapping)
- Two flavors of determining unmapping
  - Hard: make a decision as to which symbol the received signal most closely matches.
  - Soft: Assign weights to all symbols based on their respective likelihood given received signal
- Viterbi implementation we present assumes a hard decision model.



Viterbi Decoding (Overview)

- The Viterbi algorithm:
  - Given a sequence of received symbols, (that were produced by a convolution encoder, sent over a channel)
  - Determine what the input to the convolution encoder was
- It does this by determining the most likely path through the trellis

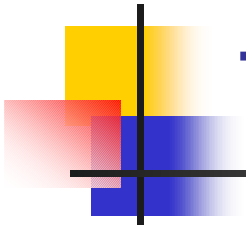
Viterbi Decoding (Main Idea)

- Dynamic Programming
  - Keep a table **c**[*s*,*t*] that records the number of errors* that would have been accumulated if the encoder was in state *s* at time *t*.
  - Also keep a table **p**[*s*,*t*] which records the state that the encoder would have been in at time *t-1* if it were in state *s* at time *t.*

* Typically calculated using either Hamming or Euclidian distance

Filling out the tables

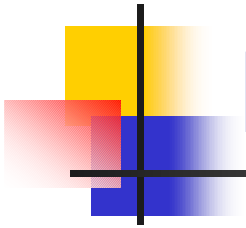
- At time t, we receive the symbol R_t.
- For each state *s*,
  - Let q_0 and q_1 be the two possible previous states of s at time t-1.
  - Let e_0 be the error between R_t and $q_0 \longrightarrow$ s
  - Let e_1 be the error between R_t and $q_1 \longrightarrow$ s
  - c[s,t]$\leftarrow$ min(**c**[q_0,*t-1*]+e_0, **c**[q_1,*t-1*]+e_1)
  - Update **p**[s,*t*] appropriately with q_0 or q_1



Traceback

- When the algorithm has examined T input symbols, it looks for the minimum entry among all states in $\mathbf{c}[s,T]$.

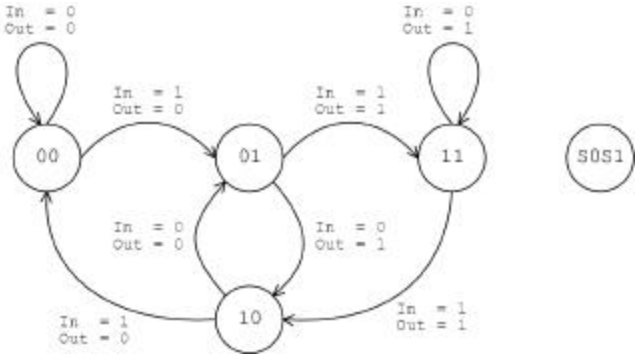
- Then the algorithm traces back through the trellis using the entries of $\mathbf{p}[s,t]$.

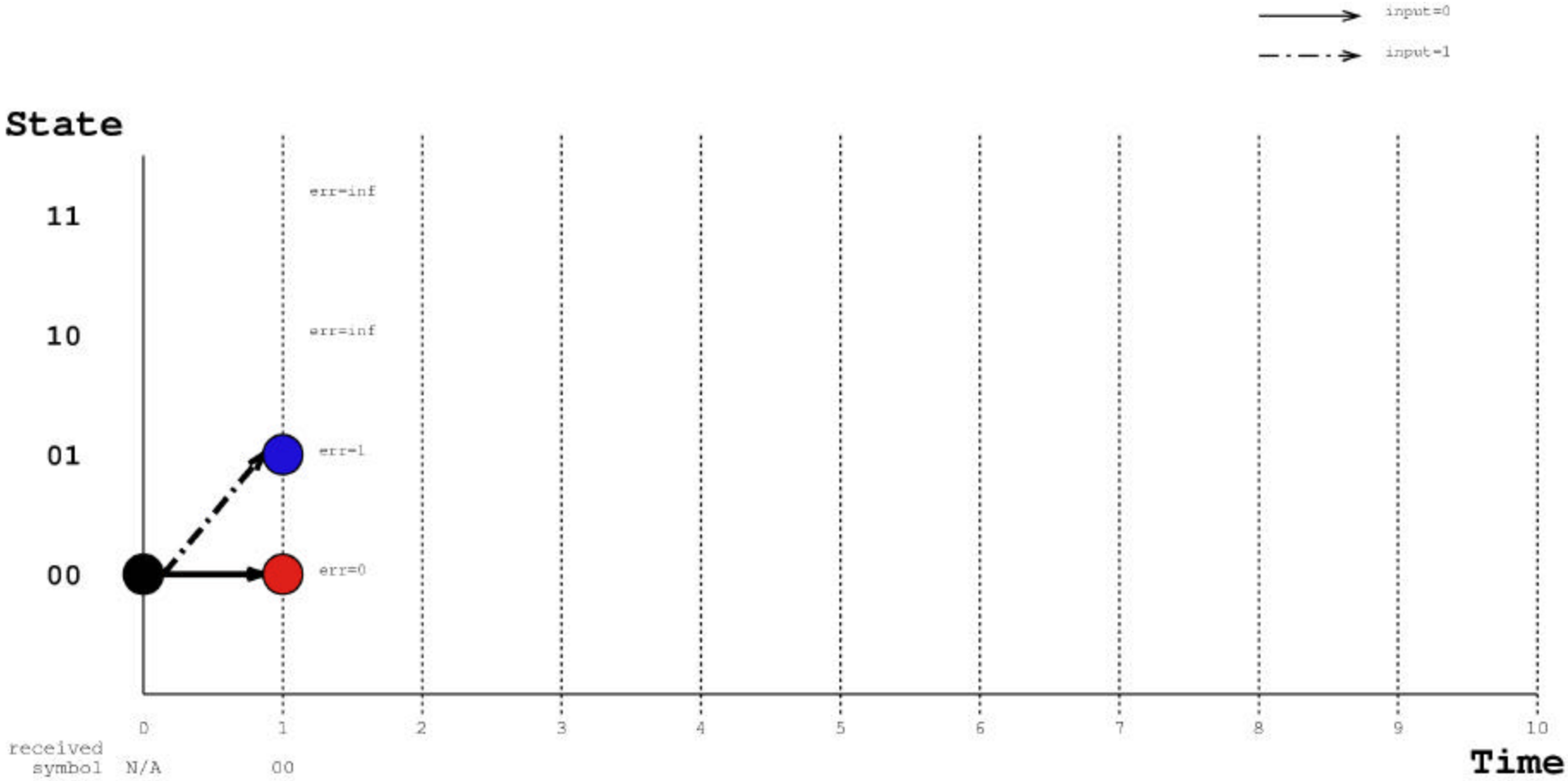


Example

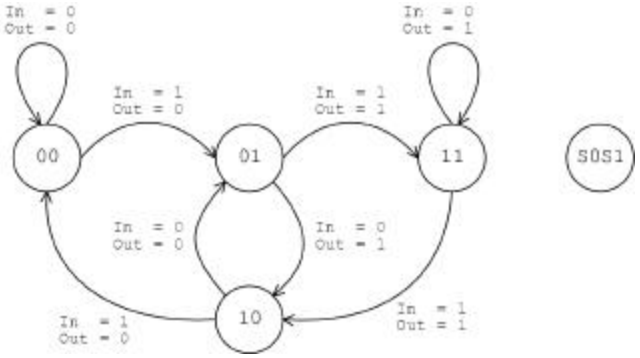
- Input to Encoder:
  - 0110101110
- Output of Encoder:
  - 00 10 11 01 11 00 11 11 10 00

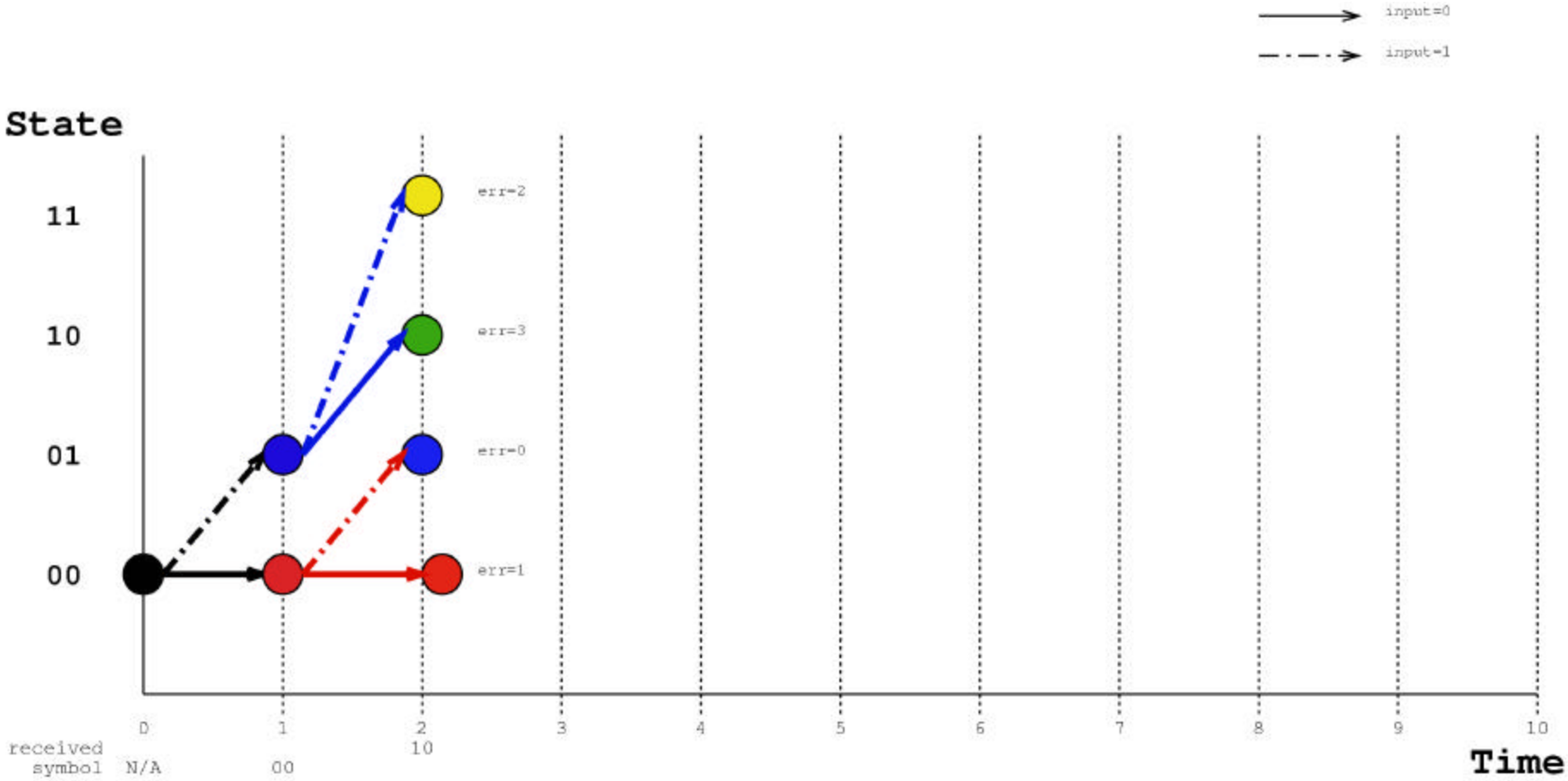
Decoder, time = 1



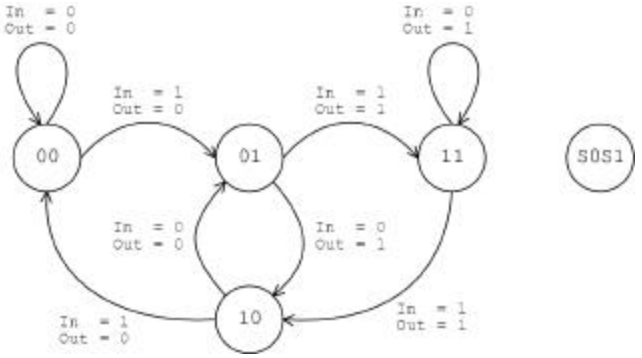


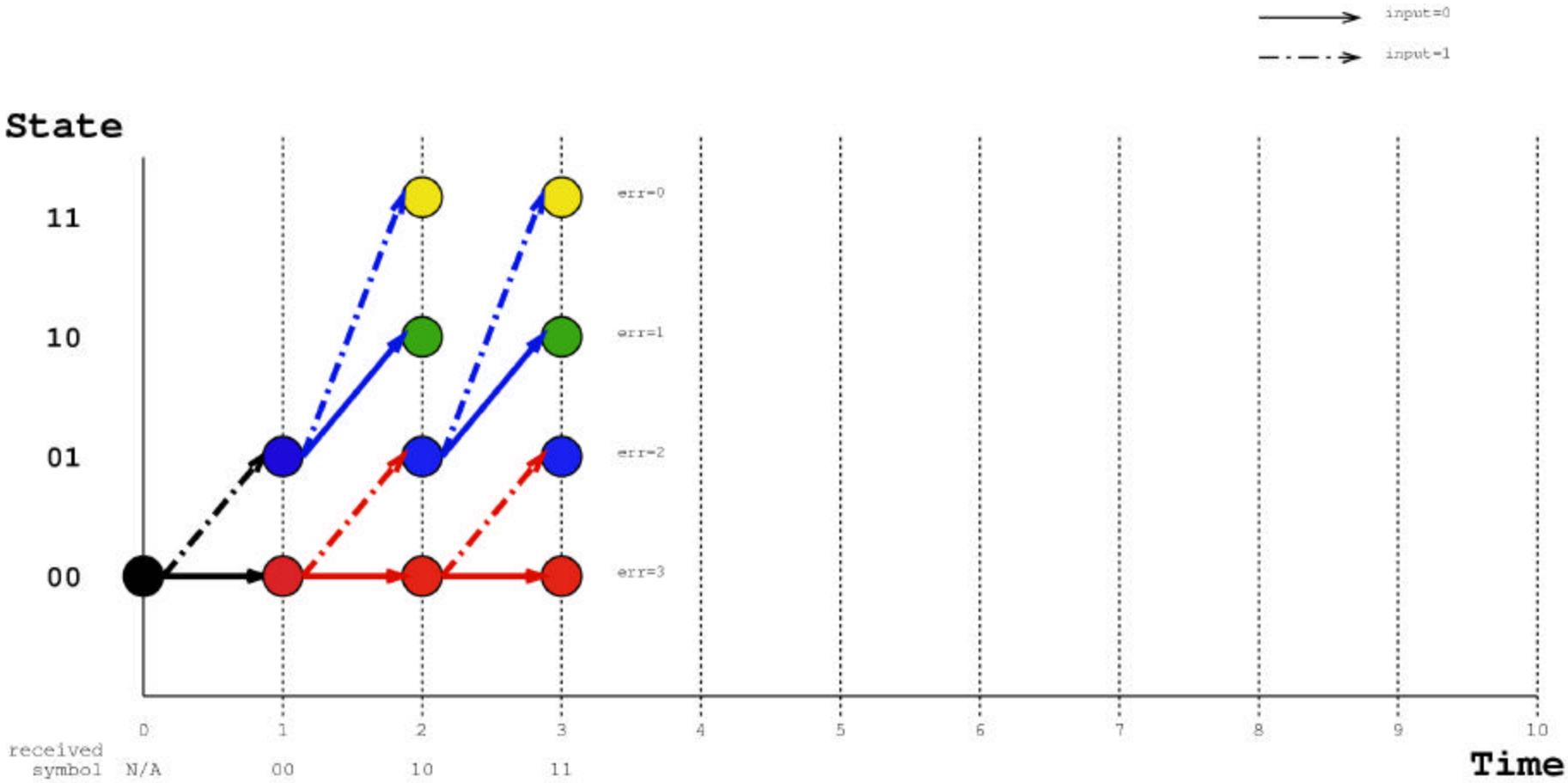
Decoder, time = 2



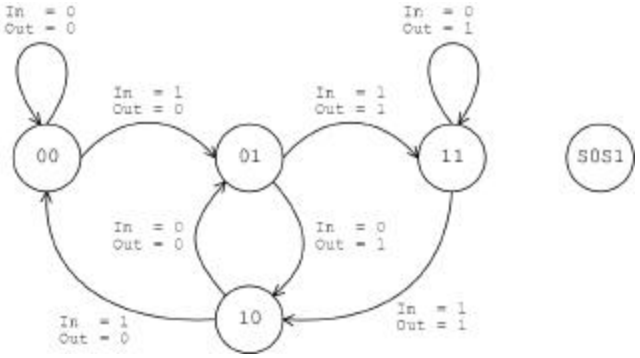


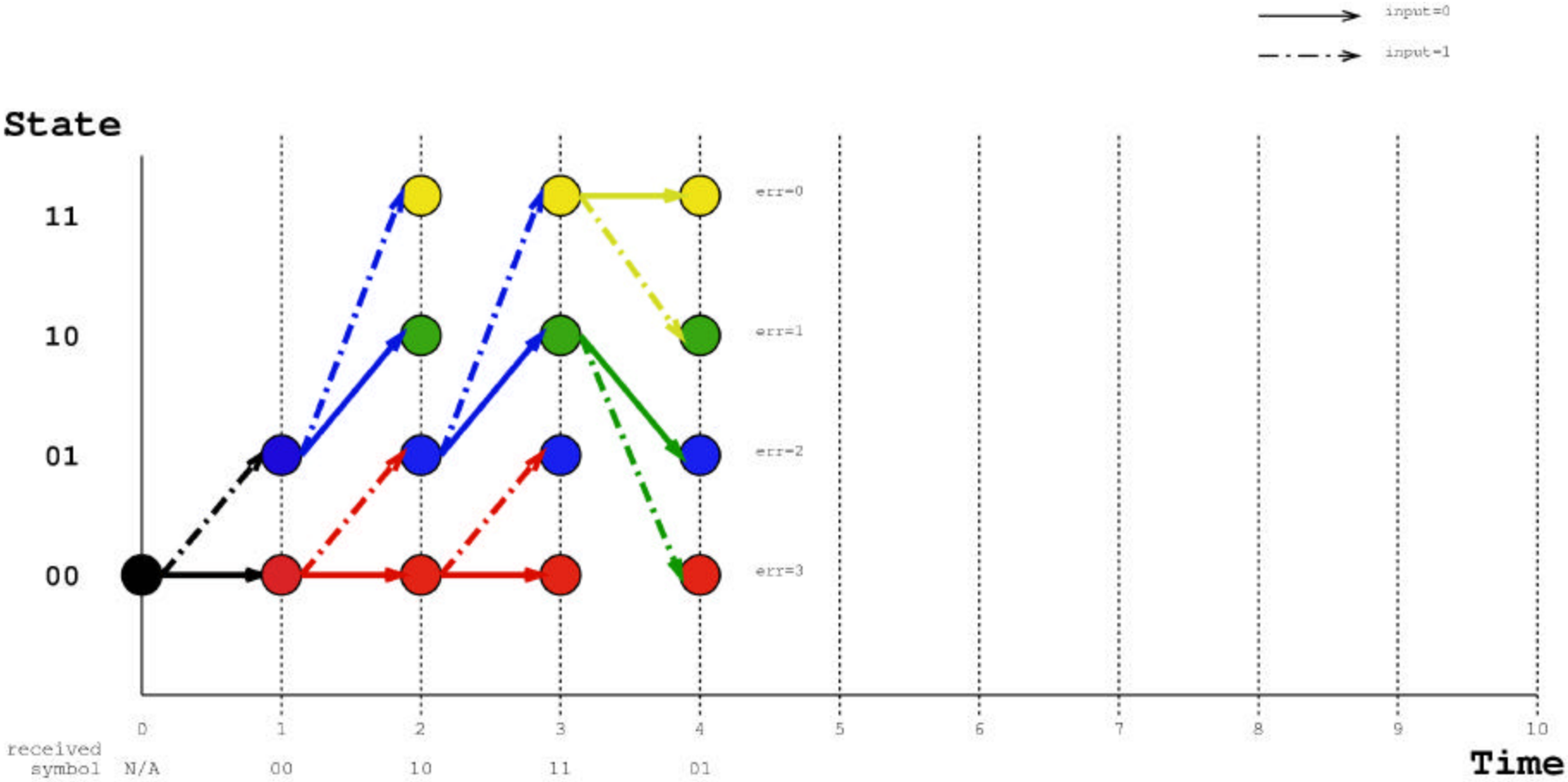
Decoder, time = 3



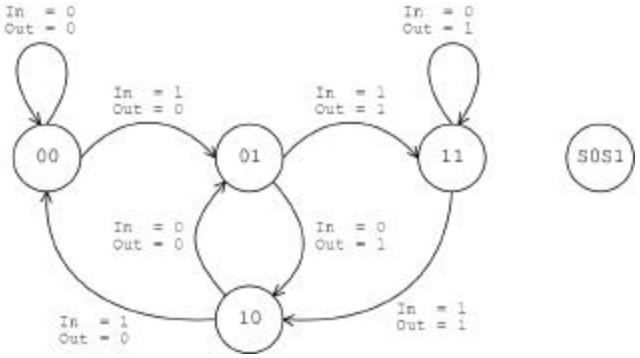


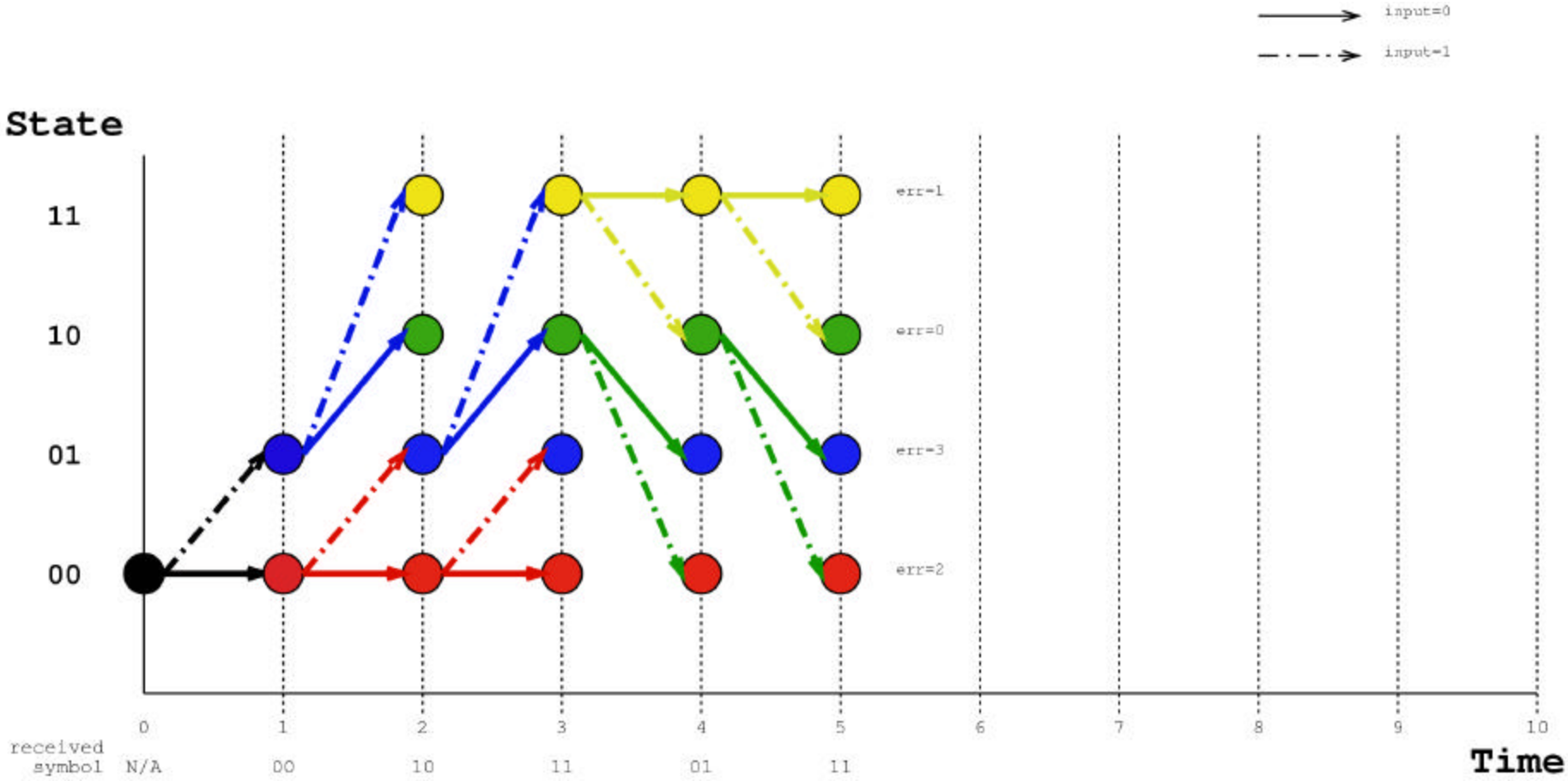
Decoder, time = 4



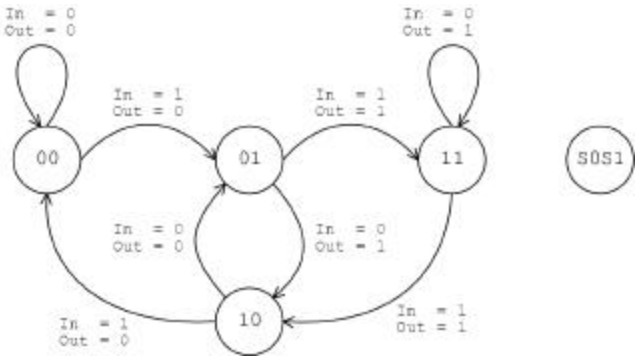


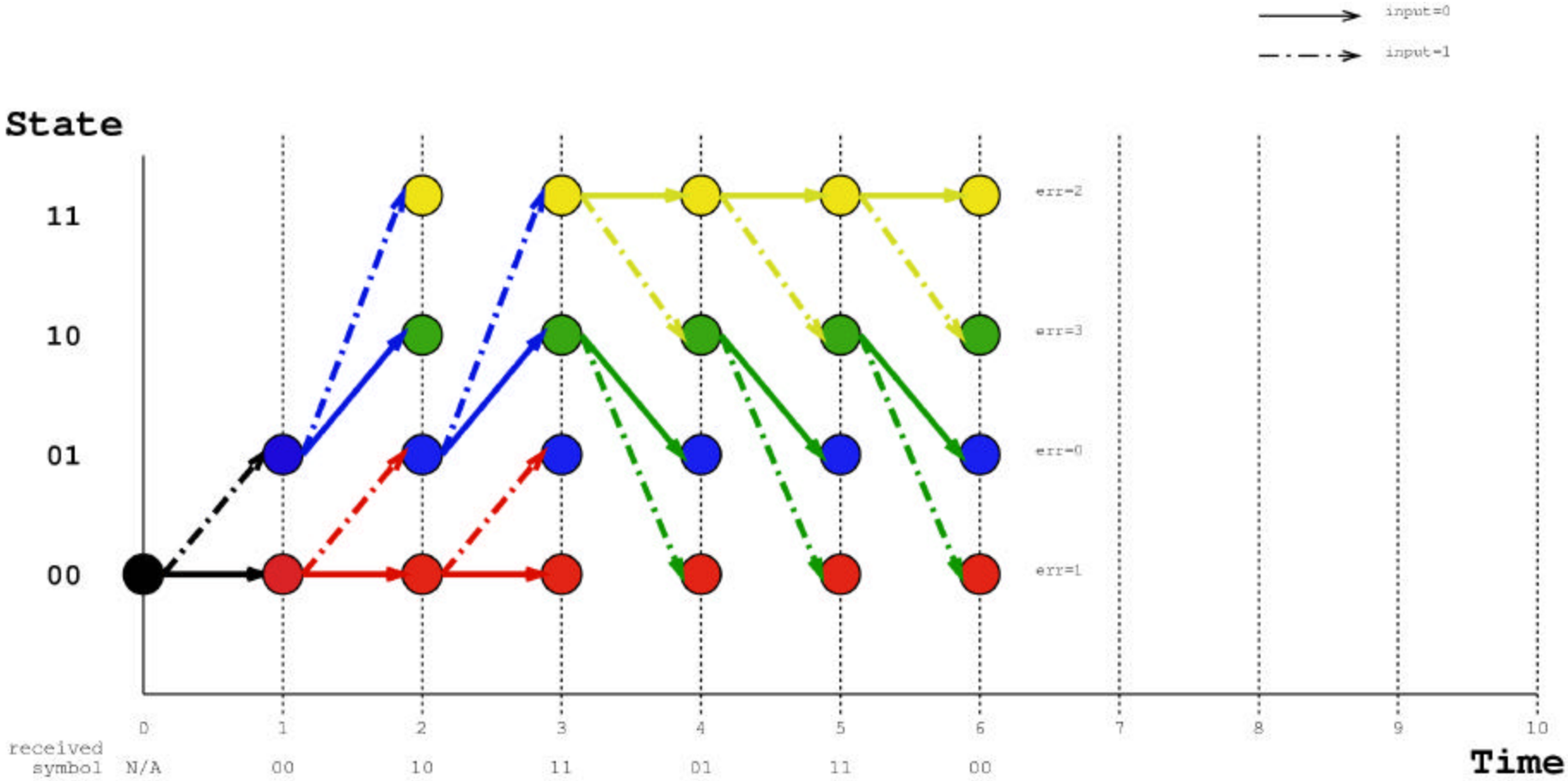
Decoder, time = 5



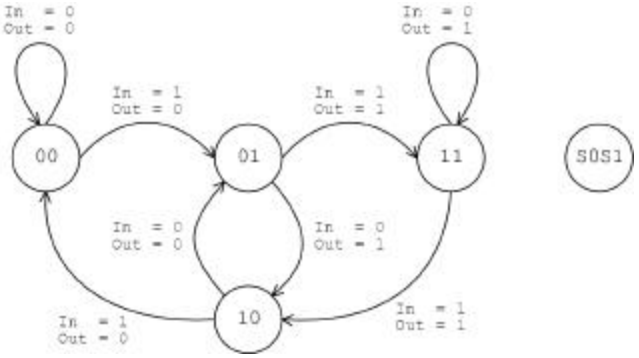


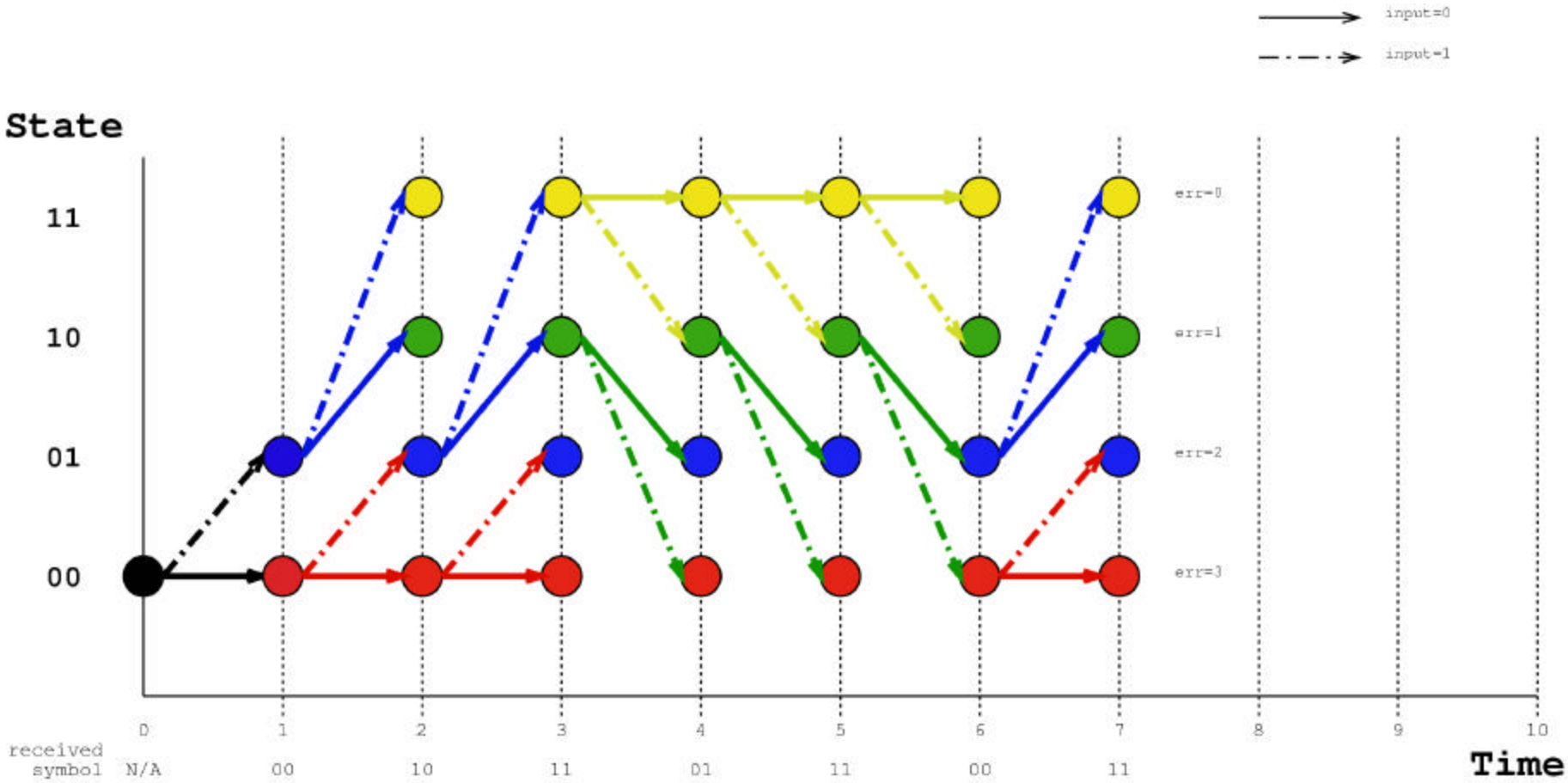
Decoder, time = 6



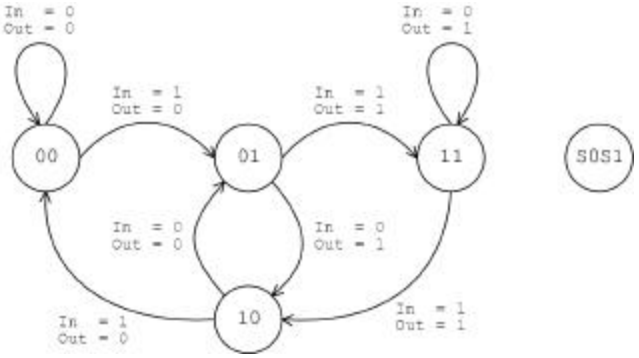


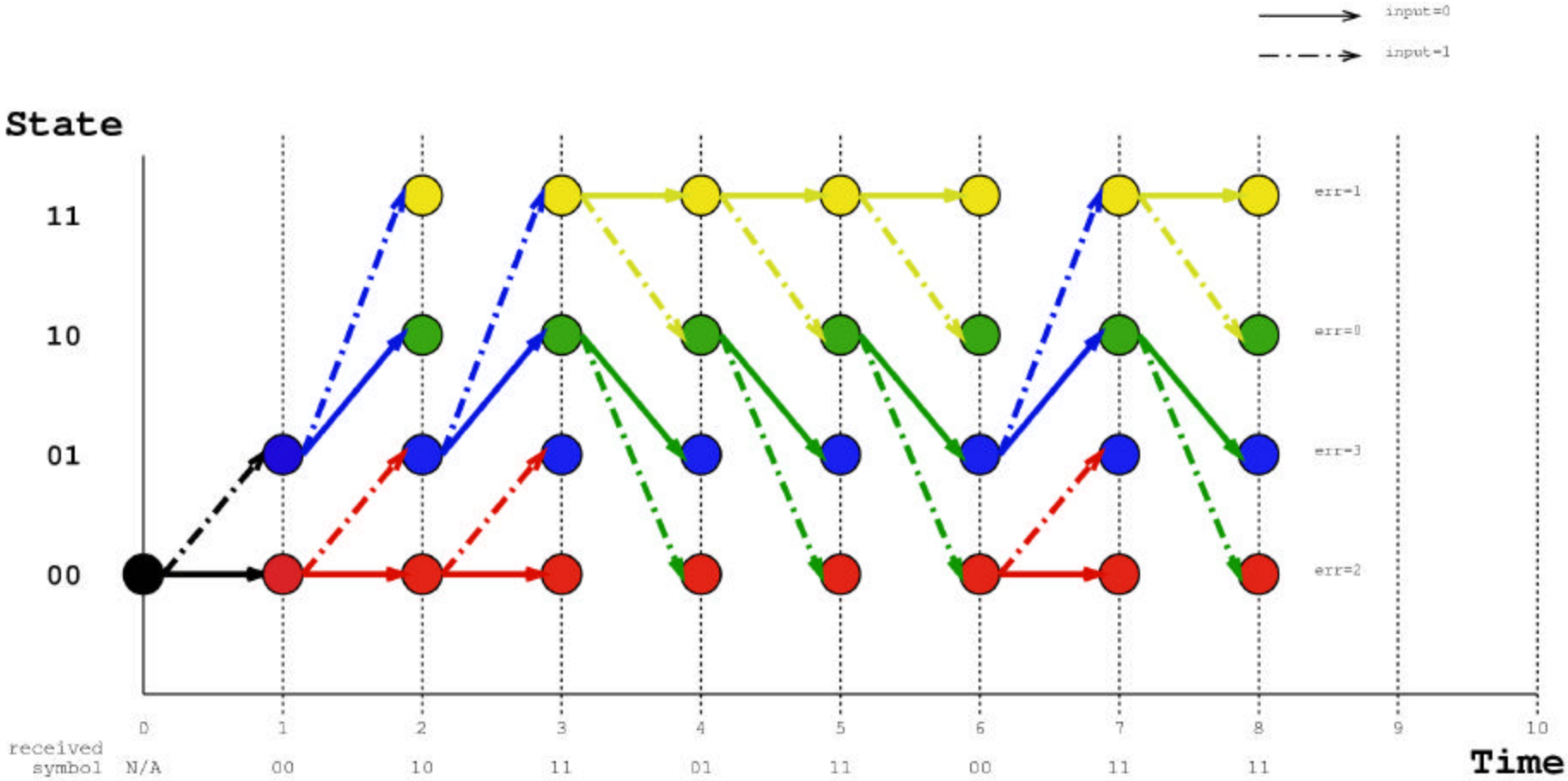
Decoder, time = 7



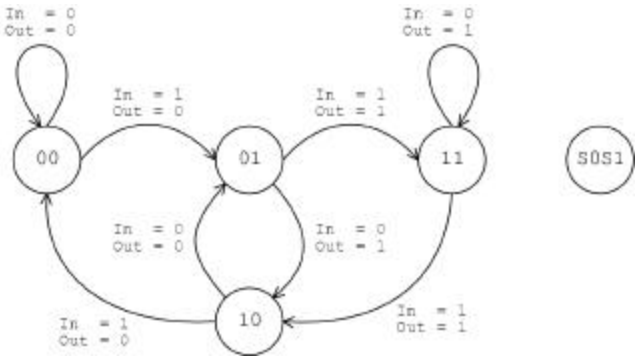


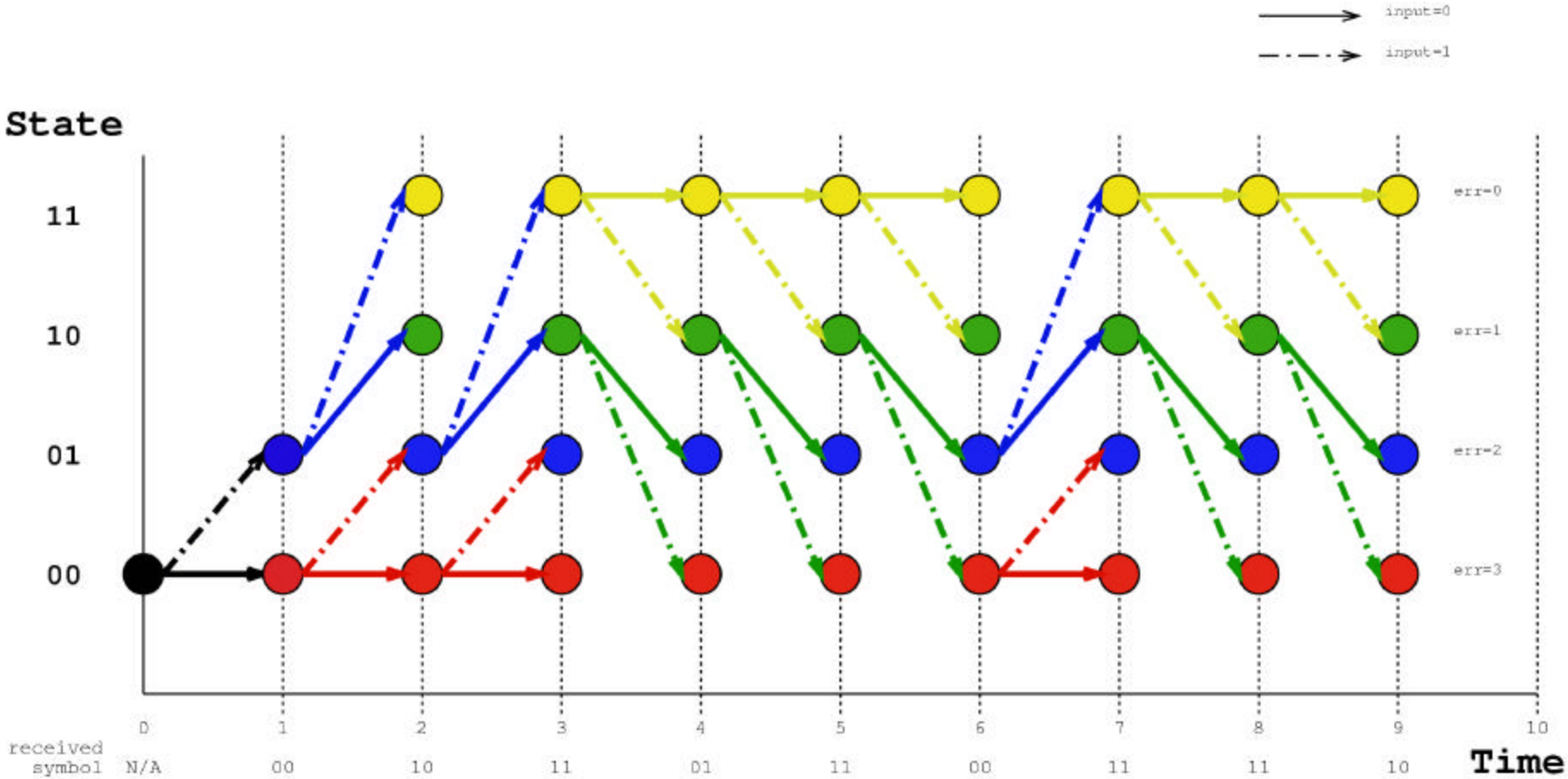
Decoder, time = 8



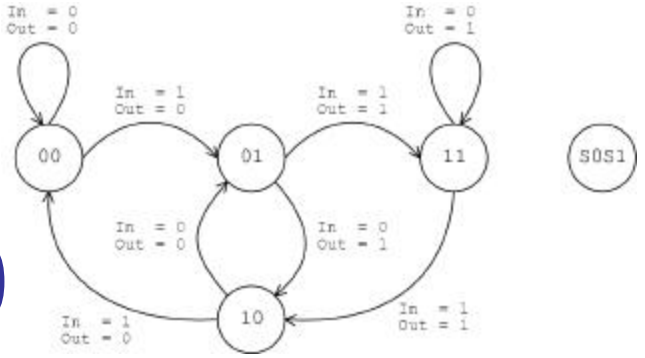


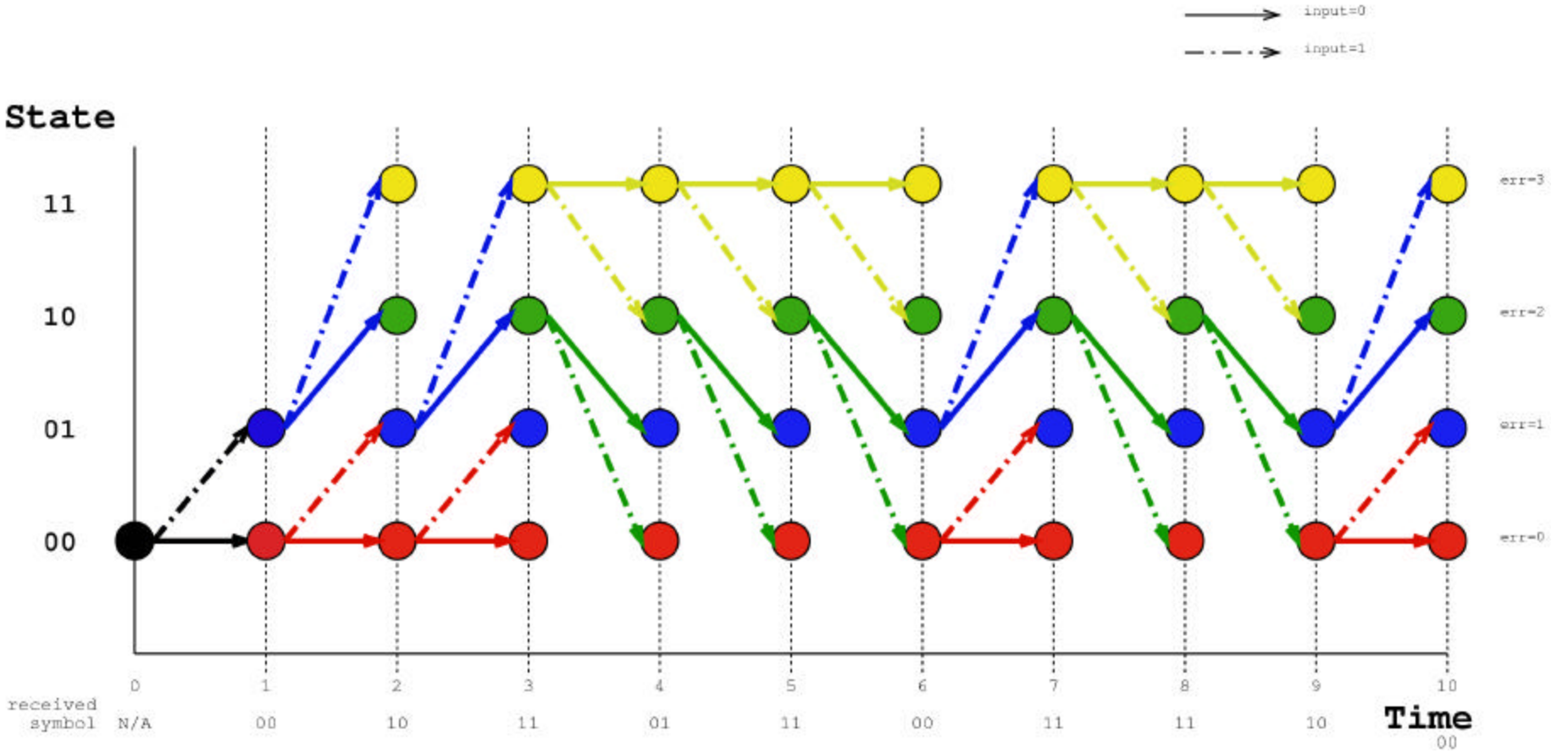
Decoder, time = 9



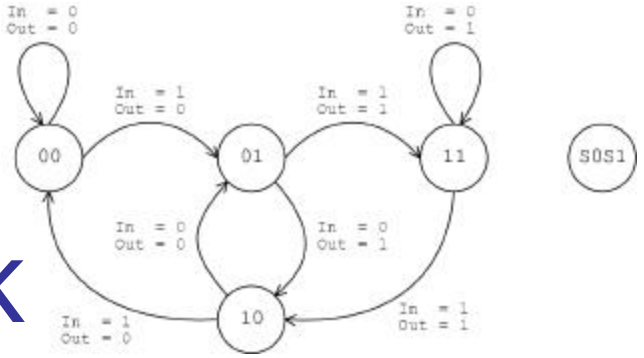


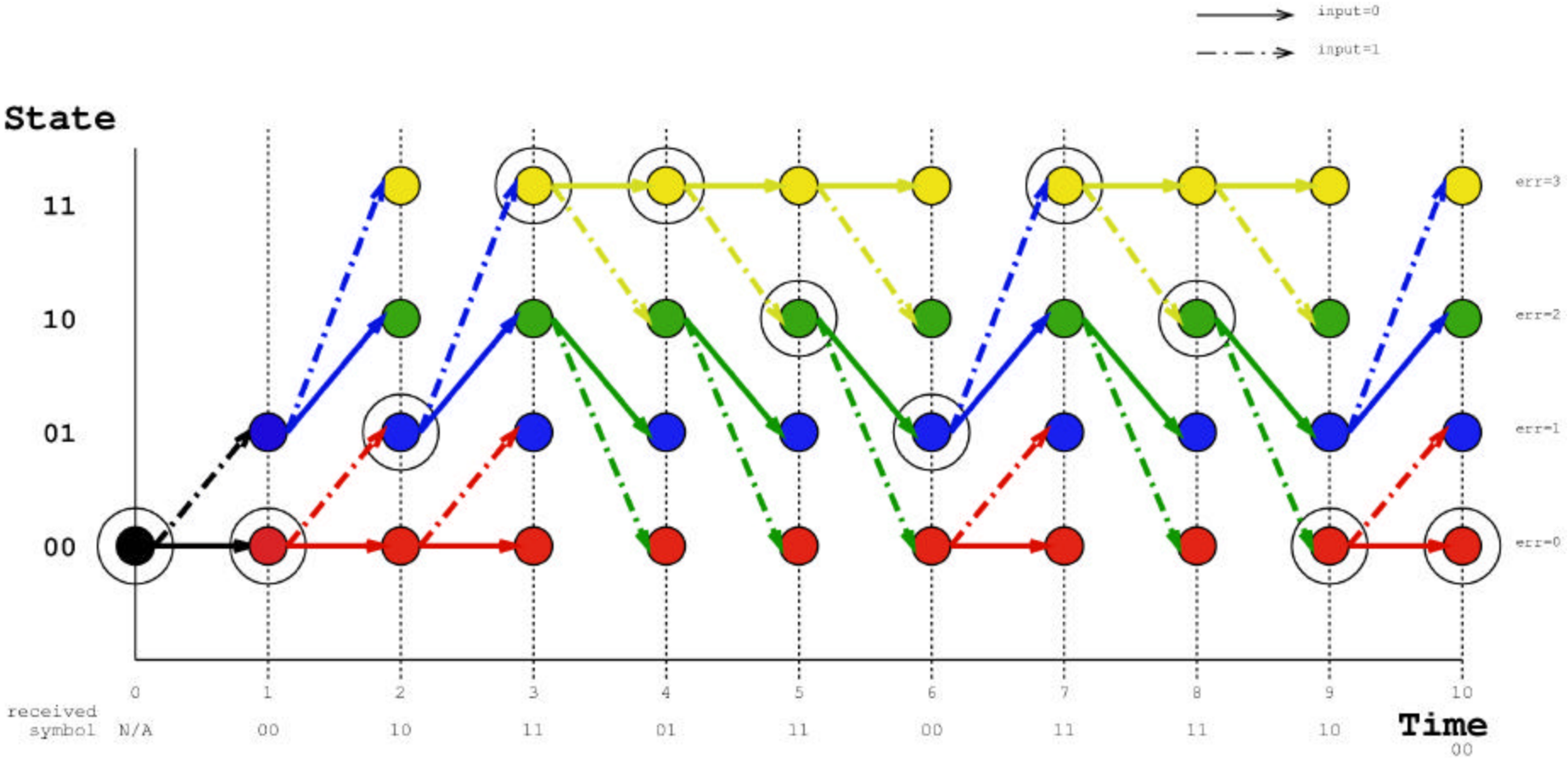
Decoder, time = 10



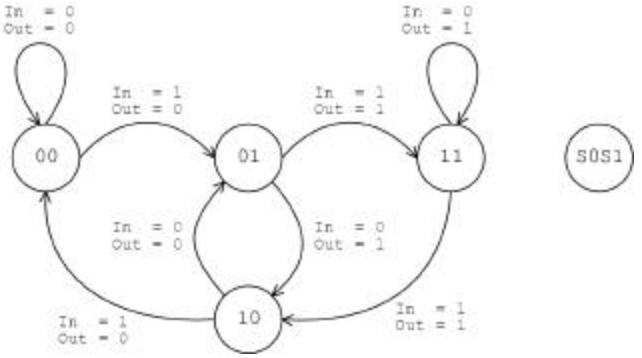


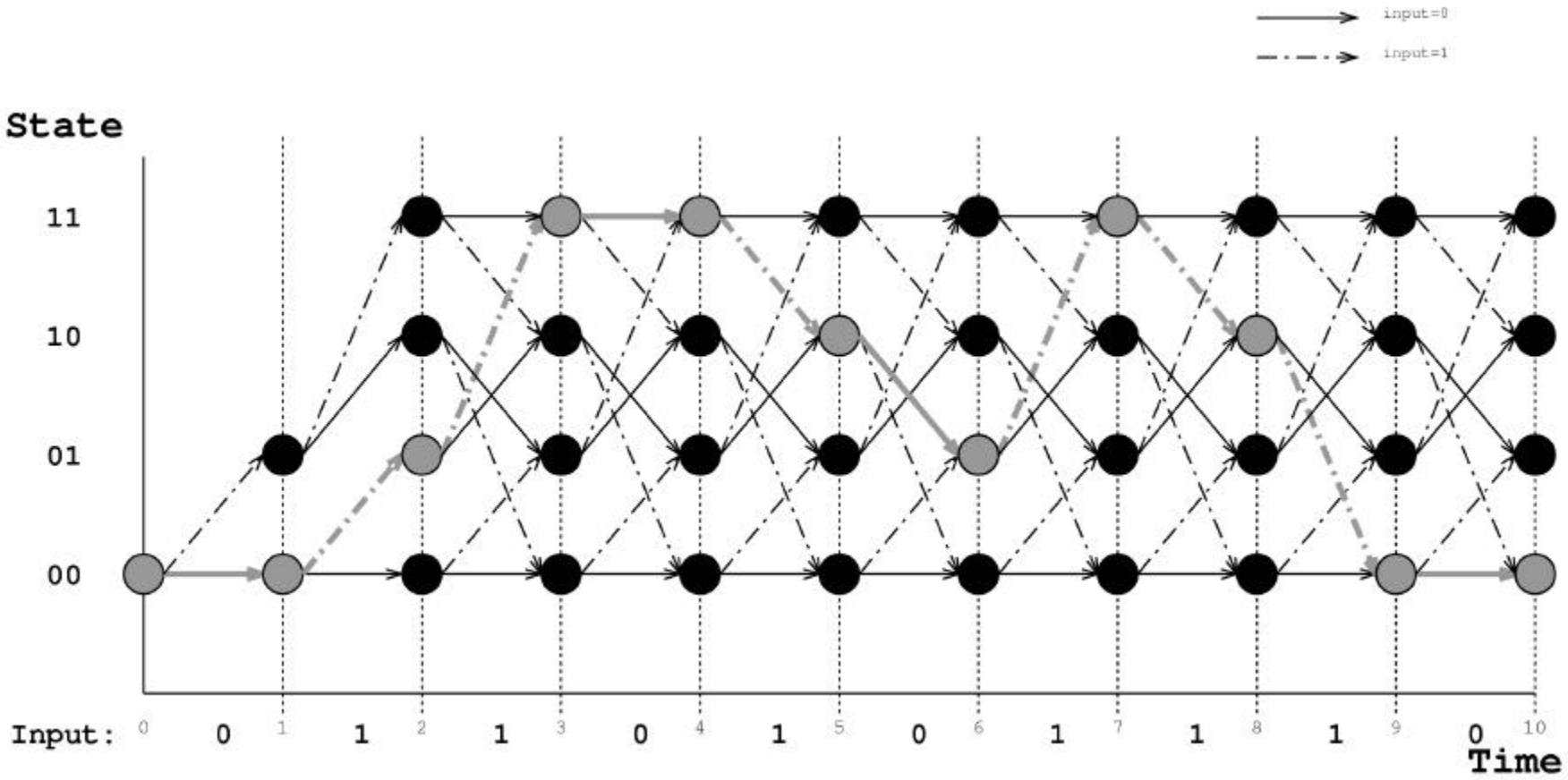
Decoder, Traceback

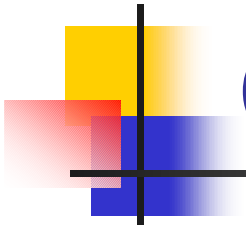




Original Path







Questions

- Is there a way to express the Viterbi algorithm in a fine grained stream graph?
- Variable rates for this decoder would mean pop(2) for 5K times, and then push(5K). This is deterministic, if not constant rate.
- Perhaps 2 stage filters?