

The states A, B, C for input 0 produce next states B, C, B, respectively, and produce next states H, G, F for input 1. The states BCB belong to the same set, but H and G, F belong to different sets. So, the set (ABC) is partitioned into (A) and (BC).

The states B and H produce next states C and D, respectively, for input 0 and produce next states G and A for input 1. C and D belong to different sets, and so the set (BH) is divided into (B) and (H).

The states E, F, G produce next states B, B, and C for input 0 and next states C, B, B for input 1. Both B and C belong to the same set, and so the set (EFG) cannot be partitioned. The new partition becomes

$$P_3 = (((A)(BC))(((D)(H))(EFG)))$$

Here, A and B are 3-distinguishable, because they produce different outputs for input string length 3. By this process, we will get P_4 also as

$$P_4 = (((A)(BC))(((D)(H))(EFG)))$$

As P_3 and P_4 consist of the same partitions,

$$P_3 = (((A)(BC))(((D)(H))(EFG)))$$

is the equivalent partition for the machine M.

Minimization: We know that equivalent partition is unique. So, $P_3 = (((A)(BC))(((D)(H))(EFG)))$ is the unique combination. Here, every single set represents one state of the minimized machine.

Let us rename these partitions for simplification.

Rename (A) as S_1 , (BC) as S_2 , (D) as S_3 , (H) as S_4 , and (EFG) as S_5 . (A) with input 0 goes to (B), and so there will be a transaction from S_1 to S_2 with input 0. (A) with input 1 goes to (H), and so there will be a transaction from S_1 to S_4 with input 1. (BC) with input 0 goes to (BC) for input 0. There will be a transaction from S_2 to S_2 for input 0. (BC) with input 1 goes to (FG). There will be a transaction from S_2 to S_5 for input 1.

By this process, the whole table of the minimized machine is constructed.

The minimized machine becomes

| Present State | Next State, z | |
|---------------|---------------|---------|
| | $X = 0$ | $X = 1$ |
| $S_1(A)$ | S_20 | S_41 |
| $S_2(BC)$ | S_20 | S_51 |
| $S_3(D)$ | S_51 | S_21 |
| $S_4(H)$ | S_31 | S_11 |
| $S_5(EFG)$ | S_21 | S_21 |

4.5 Incompletely Specified Machine, Minimal Machine

In real life, for all states and for all inputs, the next state or outputs or both are not mentioned. Those types of machines, where for all states and for all inputs, the next state, or output, or both are not mentioned, are called incompletely specified machine.

In the following machine, for state A and for 00 input, no next state and outputs are specified. So, the previous machine is an example of an incompletely specified machine.

| Present State | 00 | 01 | 11 | 10 |
|---------------|------|------|------|------|
| A | -, - | B, 0 | C, 1 | -, - |
| B | A, 0 | B, 1 | -, 0 | C, - |
| C | D, 1 | A, - | B, 0 | D, 1 |
| D | -, 0 | -, - | C, 0 | B, 1 |

4.5.1 Simplification

An incompletely specified machine can be simplified by the following steps:

- O If the next state is not mentioned for a state, for a given input, put a temporary state T in that place.
- O If the output is not mentioned, make it blank.
- O If the next state and output are not mentioned, put a temporary state T in the place of the next state and nothing in the place of output.
- O Add the temporary state T in the present state column, putting T as the next state and no output for all inputs.

By following the previous steps, the simplification of the previous incompletely specified machine will be

| Next State, z | | | | |
|---------------|------|------|------|------|
| Present State | 00 | 01 | 11 | 10 |
| A | T, - | B, 0 | C, 1 | T, - |
| B | A, 0 | B, 1 | T, 0 | C, - |
| C | D, 1 | A, - | B, 0 | D, 1 |
| D | T, 0 | T, - | C, 0 | B, 1 |
| T | T, - | T, - | T, - | T, - |

Here 'T' is the same as the dead state in finite automata.

Example 4.7 Simplify the following incompletely specified machine.

Solution:

| Next State, z | | | |
|---------------|-------|-------|-------|
| Present State | I_1 | I_2 | I_3 |
| A | C,0 | E,1 | - - |
| B | C,0 | E,- | - - |
| C | B,- | C,0 | A,- |
| D | B,0 | C,- | E,- |
| E | - - | E,0 | A,- |

Put a temporary state T in the place of the next state, where the next states are not specified. If the output is not mentioned, there is no need to put any output.

As the temporary state T is considered, put T in the present state column with the next state T for all inputs with no output.

The simplified machine becomes

| Next State, z | | | |
|---------------|-------|-------|-------|
| Present State | I_1 | I_2 | I_3 |
| A | C,0 | E,1 | T,- |
| B | C,0 | E,- | T,- |
| C | B,- | C,0 | A,- |
| D | B,0 | C,- | E,- |
| E | T,- | E,0 | A,- |
| T | T,- | T,- | T,- |

Minimal Machine: Is the minimum of the machines obtained by minimizing an incompletely specified machine.

In an incompletely specified machine, for all states and for all inputs, the next state, or output, or both are not mentioned. At the time of minimizing the incompletely specified machine, different persons can take the unmentioned next states or outputs according to their choice. Therefore, there is a great possibility to get different equivalent partitions for a single machine. But, we know that equivalent partition is unique for a given machine. It is not possible to find a unique minimized machine for a given incompletely specified machine most of the times. Therefore, our aim must be to find a reduced machine which not only covers the original machine but also has a

minimal (least of the minimum) number of states. This type of machine is called minimal machine, i.e., it is the minimum of the machines obtained by minimizing an incompletely specified machine.

Let us consider the following incompletely specified machine

| Next State, z | | |
|---------------|---------|---------|
| Present State | $X = 0$ | $X = 1$ |
| A | E,1 | D,0 |
| B | E,0 | C,1 |
| C | A,0 | B,- |
| D | A,0 | D,1 |
| E | A,- | B,0 |

In this machine, for C with input 1, the output is not specified and the same for E with input 0. There are two types of outputs that can occur in the machine. So, the unspecified outputs can be any of the following:

- (a) (B, 0 A, 0)
- (b) (B, 0 A, 1)
- (c) (B, 1 A, 1)
- (c) (B, 1 A, 0).

If it is (a), then the machine and its equivalent partition is

$$P_0 = (ABCDE)$$

$$P_1 = (A)(BD)(CE)$$

$$P_2 = (A)(B)(D)(CE)$$

| Next State, z | | |
|---------------|---------|---------|
| Present State | $X = 0$ | $X = 1$ |
| A | E,1 | D,0 |
| B | E,0 | C,1 |
| C | A,0 | B,0 |
| D | A,0 | D,1 |
| E | A,0 | B,0 |

If it is (b), then the machine and its equivalent partition is

$$P_0 = (ABCDE)$$

$$\begin{aligned}
P_1 &= (AE)(BD)(C) \\
P_2 &= (AE)(B)(D)(C) \\
P_3 &= (A)(E)(B)(D)(C)
\end{aligned}$$

| Next State, z | | |
|---------------|---------|---------|
| Present State | $X = 0$ | $X = 1$ |
| A | E,1 | D,0 |
| B | E,0 | C,1 |
| C | A,0 | B,0 |
| D | A,0 | D,1 |
| E | A,1 | B,0 |

If it is (c), then the machine and its equivalent partition is

$$\begin{aligned}
P_0 &= (ABCDE) \\
P_1 &= (AE)(BCD)
\end{aligned}$$

| Next State, z | | |
|---------------|---------|---------|
| Present State | $X = 0$ | $X = 1$ |
| A | E,1 | D,0 |
| B | E,0 | C,1 |
| C | A,0 | B,1 |
| D | A,0 | D,1 |
| E | A,1 | B,0 |

If it is (d), then the machine and its equivalent partition is

$$\begin{aligned}
P_0 &= (ABCDE) \\
P_1 &= (A)(BCD)(E) \\
P_2 &= (A)(B)(CD)(E) \\
P_3 &= (A)(B)(C)(D)(E)
\end{aligned}$$

| Next State, z | | |
|---------------|---------|---------|
| Present State | $X = 0$ | $X = 1$ |
| A | E,1 | D,0 |
| B | E,0 | C,1 |
| C | A,0 | B,1 |
| D | A,0 | D,1 |
| E | A,0 | B,0 |