The corresponding data trace of Subprocedure Branch-Trace, for \( Y \in \mathcal{L} \) is:

\[
\begin{align*}
(0, 0, 0) &= x \\
(0, 1, 0) &= x \\
(0, 1, 1) &= x \\
(0, 1, 1, 0) &= x \\
(0, 1, 1, 1) &= x \\
(0, 1, 1, 1, 0) &= x \\
(0, 1, 1, 1, 1) &= x
\end{align*}
\]

The accumulated run \( s \) and string \( x \), which are thus handled like stacks, line 4 how the new state \( s' \) and input \( i \) are pushed onto the front of \( s \). Note in line 2, \( Y \in \mathcal{L} \) is done by the nested for loops of lines 1, 2, and 3. Note in sequence for the initial pair of states \( s, s' \) of machine \( M_1 \) and \( s, s' \) of machine

Then Subprocedure Branch-Trace is called, which works backward through the provided sets \( N_{new}, \) and \( N_0 \) with product machine output label \( o \). The third pass through the do-while loop, we encounter the case \( Y(e, x) = 0 \). On the fourth pass through do-while loop, we look at all transitions out of

\[
\begin{align*}
\{(s, f) \} &= \emptyset \quad N_{new} \quad Y(e, x) \quad N_0 \quad o \quad Y(e, x) \quad N_{new} \quad Y(e, x) \quad N_0 \quad o
\end{align*}
\]

Figure 7.43: Procedure for finding a shortest error trace.

```plaintext
procedure error-trace(s, x, e, f)
\}
```

Chapter 7. Models of Sequential Systems
Another example of equivalence checking based on the product operation.

**Example:**

This is which takes the initial state to the other transition.

Procedure 7.4: Procedure for equivalence checking a product machine.

```plaintext
{ 
  return(TRUE)
}

with

while (New ≠ Ø) {
  Reachable = Reachable ∩ New
  from
  New = New
}

{ 
  return(FALSE)
}

error-trace(S) = ((x, y, z), 0, 0, ..., 0) = 0
```

Figure 7.4.2: Procedure for equivalence checking a product machine.

1. Initialize Time = 0, From = new, S0 = Ø
2. while (Reachable ≠ From) do
3. 
4. for s ∈ S
5. 
6. while (New ≠ Ø) do
7. 
8. for x ∈ X
9. 
10. for y ∈ Y
11. 
12. for z ∈ Z
13. 
14. return(TRUE)
15. 
16. return(FALSE)