A1 Q4

Zhengyang Fei

First we write function for all the cases according to diagram.

```r
# Function to compute P(A)
p_a <- function(a) {
  if (a == 1) {
    return(0.01)  # P(A = 1)
  } else {
    return(0.99)  # P(A = 0)
  }
}

# P(T | A)
p_ta <- function(t, a) {
  if (a == 1) {
    p <- 0.05  # P(T = 1 | A = 1)
  } else if (a == 0) {
    p <- 0.01  # P(T = 1 | A = 0)
  }
  return(ifelse(t == 1, p, 1.0 - p))
}

# P(X | E)
p_xe <- function(x, e) {
  if (e == 1) {
    p <- 0.98  # P(X = 1 | E = 1)
  } else if (e == 0) {
    p <- 0.05  # P(X = 1 | E = 0)
  }
  return(ifelse(x == 1, p, 1.0 - p))
}

# P(E | T, L)
```

```r
p_tl <- function(e, t, l) {
  if (t == 1 && l == 1) {
    p <- 1.0  # P(E = 1 | T = 1, L = 1)
  } else if (t == 1 && l == 0) {
    p <- 1.0  # P(E = 1 | T = 1, L = 0)
  } else if (t == 0 && l == 1) {
    p <- 1.0  # P(E = 1 | T = 0, L = 1)
  } else if (t == 0 && l == 0) {
    p <- 0.0  # P(E = 0 | T = 0, L = 0)
  }
  return(ifelse(e == 1, p, 1.0 - p))
}

# P(D | E, B)
p_eb <- function(d, e, b) {
  if (e == 1 && b == 1) {
    p <- 0.90  # P(D = 1 | E = 1, B = 1)
  } else if (e == 1 && b == 0) {
    p <- 0.70  # P(D = 1 | E = 1, B = 0)
  } else if (e == 0 && b == 1) {
    p <- 0.80  # P(D = 1 | E = 0, B = 1)
  } else if (e == 0 && b == 0) {
    p <- 0.10  # P(D = 1 | E = 0, B = 0)
  }
  return(ifelse(d == 1, p, 1.0 - p))
}

# P(L | S)
p_ls <- function(l, s) {
  if (s == 1) {
    p <- 0.10  # P(L = 1 | S = 1)
  } else if (s == 0) {
    p <- 0.01  # P(L = 1 | S = 0)
  }
  return(ifelse(l == 1, p, 1.0 - p))
}

# P(B | S)
p_bs <- function(b, s) {
  if (s == 1) {
    p <- 0.30  # P(B = 1 | S = 1)
  } else if (s == 0) {
```

```
    p <- 0.60  # P(B = 1 | S = 0)
  }
  return(ifelse(b == 1, p, 1.0 - p))
}

# P(S)
p_s <- function(s) {
  if (s == 1) {
    return(0.50)  # P(S = 1)
  } else {
    return(0.50)  # P(S = 0)
  }
}
```

Part A

```
## Question 4 Part a
# Generate all combinations of A, T, X, E, D, L, B, S (0 and 1 for each)
# variable mat is a 256 by 8 matrix
mat <- expand.grid(A = 0:1, T = 0:1, X = 0:1, E = 0:1, D = 0:1, L = 0:1, B = 0:1, S = 0:1)

prob_sum <- 0

# Loop through each combination in mat
for (i in 1:nrow(mat)) {
  row <- mat[i, ]

  p <- p_a(row$A) *
    p_ta(row$T, row$A) *
    p_xe(row$X, row$E) *
    p_tl(row$E, row$T, row$L) *
    p_eb(row$D, row$E, row$B) *
    p_ls(row$L, row$S) *
    p_bs(row$B, row$S) *
    p_s(row$S)

  # Add to the sum of joint probabilities
  prob_sum <- prob_sum + p
}
```

Summing over all possible combinations of variables, we see that

```
print(prob_sum)
```

```
[1] 1
```

Part B

```r
## Question 4 Part b
# Initialize variables for P(L = 1), P(L = 1 | A = 1), and P(L = 1 | A = 0)
P_L1 <- 0
P_L1_given_A1 <- 0
P_L1_given_A0 <- 0
P_A1 <- 0
P_A0 <- 0

for (i in 1:nrow(mat)) {
  row <- mat[i, ]

  p <- p_a(row$A) *
    p_ta(row$T, row$A) *
    p_xe(row$X, row$E) *
    p_tl(row$E, row$T, row$L) *
    p_eb(row$D, row$E, row$B) *
    p_ls(row$L, row$S) *
    p_bs(row$B, row$S) *
    p_s(row$S)

  # Add to P(L = 1) if L = 1
  if (row$L == 1) {
    P_L1 <- P_L1 + p
  }

  # Add to P(L = 1 | A = 1) if A = 1 and L = 1
  if (row$A == 1 && row$L == 1) {
    P_L1_given_A1 <- P_L1_given_A1 + p
  }

  # Add to P(L = 1 | A = 0) if A = 0 and L = 1
  if (row$A == 0 && row$L == 1) {
    P_L1_given_A0 <- P_L1_given_A0 + p
  }

  # P(A = 1) and P(A = 0)
  if (row$A == 1) {
    P_A1 <- P_A1 + p
  }
```

```r
  if (row$A == 0) {
    P_A0 <- P_A0 + p
  }
}

# Applying conditional probability
P_L1_given_A1 <- P_L1_given_A1 / P_A1
P_L1_given_A0 <- P_L1_given_A0 / P_A0

# Output
print(P_L1)
```

```
[1] 0.055
```

```r
print(P_L1_given_A1)
```

```
[1] 0.055
```

```r
print(P_L1_given_A0)
```

```
[1] 0.055
```

Since $P(L = 1) = P(L = 1|A = 1) = P(L = 1|A = 0) = 0.055$, $A \perp\!\!\!\perp L$ as needed.

Part C

```
# Initialize variables for conditional probabilities
P_L1_given_X1 <- 0
P_L1_given_A1_X1 <- 0
P_L1_given_A0_X1 <- 0
P_X1 <- 0
P_A1_X1 <- 0
P_A0_X1 <- 0

# Loop through each combination and calculate the joint probabilities
for (i in 1:nrow(mat)) {
  row <- mat[i, ]

  p <- p_a(row$A) *
    p_ta(row$T, row$A) *
    p_xe(row$X, row$E) *
    p_tl(row$E, row$T, row$L) *
    p_eb(row$D, row$E, row$B) *
    p_ls(row$L, row$S) *
    p_bs(row$B, row$S) *
    p_s(row$S)

  # Add to P(L = 1 | X = 1) if L = 1 and X = 1
  if (row$L == 1 && row$X == 1) {
    P_L1_given_X1 <- P_L1_given_X1 + p
  }

  # Add to P(L = 1 | A = 1, X = 1) if L = 1, A = 1, and X = 1
  if (row$L == 1 && row$A == 1 && row$X == 1) {
    P_L1_given_A1_X1 <- P_L1_given_A1_X1 + p
  }

  # Add to P(L = 1 | A = 0, X = 1) if L = 1, A = 0, and X = 1
  if (row$L == 1 && row$A == 0 && row$X == 1) {
    P_L1_given_A0_X1 <- P_L1_given_A0_X1 + p
  }

  # Add to P(X = 1)
  if (row$X == 1) {
    P_X1 <- P_X1 + p
```

```r
  }

  # Add to P(A = 1, X = 1) if A = 1 and X = 1
  if (row$A == 1 && row$X == 1) {
    P_A1_X1 <- P_A1_X1 + p
  }

  # Add to P(A = 0, X = 1) if A = 0 and X = 1
  if (row$A == 0 && row$X == 1) {
    P_A0_X1 <- P_A0_X1 + p
  }
}

# Apply conditional probability
P_L1_given_X1 <- P_L1_given_X1 / P_X1
P_L1_given_A1_X1 <- P_L1_given_A1_X1 / P_A1_X1
P_L1_given_A0_X1 <- P_L1_given_A0_X1 / P_A0_X1

# Output the results
print(P_L1_given_X1)
```

```
[1] 0.4887114
```

```r
print(P_L1_given_A1_X1)
```

```
[1] 0.3714872
```

```r
print(P_L1_given_A0_X1)
```

```
[1] 0.4902741
```

Hence ,we see that

$$\begin{aligned} P(L = 1 \mid X = 1) &= 0.4887114 \\ P(L = 1 \mid A = 1, X = 1) &= 0.3714872 \\ P(L = 1 \mid A = 0, X = 1) &= 0.4902741 \end{aligned}$$

therefore they are not independent as they are not equal.