Tutorial 9

Part 1:

1. The basic principle being used in the Rejection-Sampling algorithm involves making use of the prior distribution specified for the network to generate samples from and then rejecting those samples that do not match with the evidence. So, for our case the evidence would be $\neg l \land r$.

d	e	a	b	r	l	
T	F	T	F	F	F	F
F	F	T	F	T	F	T
F	T	F	T	T	T	F
T	F	T	T	T	F	T
F	F	F	T	F	F	F

Summarised from the book, the main problem with using rejection sampling is that it rejects so many samples.

2. Set the weight to 1.0
Sample from P(D) = <0.75, 0.25>, suppose it returns false.
Sample from P(E) = <0.95, 0.05> suppose it returns true.
Sample from P(A|D = false, E = true) = <0.1, 0.9>, suppose it returns true.
Sample from P(B|A = true) = <0.8, 0.2>, suppose it returns true.

L is an evidence variable with value false. Therefore, we set:
w ← w × P(L = false|B = true) = 0.1
R is an evidence variable with value True. Therefore, we set:
w ← w × P(R = true|B = true) = 0.08
So we obtain a sample [false, true, true, true, false, true] with weight 0.08 under D = false.

Part 2:

1a) α is the normalisation factor.
P(e_{t+1} | x_{t+1}) is obtained from the sensor model.
P(X_{t+1} | x_t) is obtained from the transition model.
P(e_t | $x_{1:t}$) is obtained from the current state distribution and is the recursive term.

b) P(e_{k+1} | x_{k+1}) is obtained from the sensor model.
P($e_{k+2:t}$ | x_{k+1}) is the recursive term.
P(x_{k+1} | X_k) is obtained from the transition model.

2) b(k+1:t) = $P(\neg on_2|Tr_1) = \sum_{tr_2} P(\neg on_2|tr_2)P(tr_2)P(tr_2|Tr_1)\, where P(\neg on_2|tr_2) = 0.2 P(|\mathrm{tr}_2) = 1 and P(tr_2|Tr_1) is < 0.8,0.9 > for tr_2 and < 0.2,0.1 > for \neg tr_2 = (0.2 \times 1 \times < 0.8, 0.9 >) + (0.9 \times 1 \times < 0.2, 0.1 >) =< 0.16,0.18 > +< 0.18,0.09 \geq< 0.34,0.27 >.$

3) $P(Tr_1|on_1, \neg on_2) = \alpha f_1 : b_{(k+1:t)} = \alpha P\big(X_k\big|e_{(1:k)}\big)P\big(e_{(k+1:t)}\big|X_k\big) = \alpha < 0.664,\ 0.017 > \times < 0.34,\ 0.27 > = \ \alpha < 0.22576,\ 0.00459 > = \frac{1}{0.22576 + 0.00459} < 0.22576, 0.00459 > = < 0.98, 0.02 >$

4) Yes, as it has a single state variable $Train_t$ and a single evidence variable $OnTime_t$.

5) Assigning i = 1 for $X_{(t-1)}$ = true, i = 2 for $X_{(t-1)}$ = false, j = 1 for X_t = true and j = 2 for X_t = false. By using $T_{(ij)}$ = P(X_t = j | $X_{(t-1)}$ = i) we get that,

T = $\begin{pmatrix} P\big(x_t\big|x_{(t-1)}\big) & P\big(\neg x_t\big|x_{(t-1)}\big) \\ P\big(x_t\big|\neg x_{(t-1)}\big) & P(\neg x\_t|\neg x)(t-1)) \end{pmatrix}$ = $\begin{pmatrix} 0.8 & 0.2 \\ 0.9 & 0.1 \end{pmatrix}$.

6) O1 = $\begin{pmatrix} P(On_1 = true \mid Tr_1 = true) & 0 \\ 0 & P(On_1 = true \mid Tr_1 = false) \end{pmatrix}$ = $\begin{pmatrix} 0.8 & 0 \\ 0 & 0.1 \end{pmatrix}$.

O2 = $\begin{pmatrix} P(On_2 = \mathrm{false} \mid Tr_2 = true) & 0 \\ 0 & P(On_2 = \mathrm{false} \mid Tr_2 = false) \end{pmatrix}$ = $\begin{pmatrix} 0.2 & 0 \\ 0 & 0.9 \end{pmatrix}$.

7a) $f_{(1:k)} = f_{(1:1)} = \alpha O_1 T^T f_0 = \alpha \begin{pmatrix} 0.8 & 0 \\ 0 & 0.1 \end{pmatrix} \begin{pmatrix} 0.8 & 0.2 \\ 0.9 & 0.1 \end{pmatrix}^T \begin{pmatrix} 0.7 \\ 0.3 \end{pmatrix} = \begin{pmatrix} 0.8 & 0 \\ 0 & 0.1 \end{pmatrix} \begin{pmatrix} 0.8 & 0.9 \\ 0.2 & 0.1 \end{pmatrix} \begin{pmatrix} 0.7 \\ 0.3 \end{pmatrix} = \begin{pmatrix} 0.975 \\ 0.025 \end{pmatrix}$.

b) $\mathrm{b}_{(\mathrm{k}+1:\mathrm{t})} = \mathrm{b}_{(2:2)} = T\mathrm{O}_2 \mathrm{b}_{3:2} = \begin{pmatrix} 0.8 & 0.2 \\ 0.9 & 0.1 \end{pmatrix} \begin{pmatrix} 0.2 & 0 \\ 0 & 0.9 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.34 \\ 0.27 \end{pmatrix}$.

c) $P\left(X_k \middle| e_{(1:t)}\right) = \alpha f_{(1:k)} b_{(k+1:t)}$ *would become* $P(Tr_1 | on_1, \neg on_2) = \alpha \begin{pmatrix} 0.975 \\ 0.025 \end{pmatrix} \begin{pmatrix} 0.34 \\ 0.27 \end{pmatrix} = \begin{pmatrix} 0.98 \\ 0.0199 \end{pmatrix}$.