- 15.2 (Zhikui) In this question, we examine what happens to the probabilities in the umbrella world in the limit of long time sequences.

  - Suppose we observe an unending sequence of days in which the umbrella appears. Show that, as the days go by, the probability of rain the current day increases monotonically toward a fixed point. Calculate this fixed point.

  - Now consider forecasting further and further into the future, given just the first two days where *Umbrella = true*. First, compute the probability $P(r_{2+k} \mid u_1, u_2)$ for 1..5. You should see that the values converge towards a fixed point. Prove that the value of this fixed point is 0.5.

15.2

a. For all t, we have the filtering formula

$$\mathbf{P}(R_t|u_{1:t}) = \alpha \mathbf{P}(u_t|R_t) \sum_{R_{t-1}} \mathbf{P}(R_t|R_{t-1}) P(R_{t-1}|u_{1:t-1}) .$$

At the fixed point, we additionally expect that $\mathbf{P}(R_t|u_{1:t}) = \mathbf{P}(R_{t-1}|u_{1:t-1})$. Let the fixed-point probabilities be $\langle \rho,\ 1-\rho \rangle$. This provides us with a system of equations:

$$\begin{aligned}
\langle \rho,\ 1-\rho \rangle &= \alpha \langle 0.9,\ 0.2 \rangle \langle 0.7,\ 0.3 \rangle \rho + \langle 0.3,\ 0.7 \rangle (1-\rho) \\
&= \alpha \langle 0.9,\ 0.2 \rangle (\langle 0.4\rho,\ -0.4\rho \rangle + \langle 0.3,\ 0.7 \rangle) \\
&= \frac{1}{0.9(0.4\rho + 0.3) + 0.2(-0.4\rho + 0.7)} \langle 0.9,\ 0.2 \rangle (\langle 0.4\rho,\ -0.4\rho \rangle + \langle 0.3,\ 0.7 \rangle)
\end{aligned}$$

Solving this system, we find that $\rho \approx 0.8933$.

b. The probability converges to $\langle 0.5, 0.5 \rangle$ as illustrated in Figure S15.1. This convergence makes sense if we consider a fixed-point equation for $\mathbf{P}(R_{2+k}|U_1, U_2)$:

$$\begin{aligned}
\mathbf{P}(R_{2+k}|U_1, U_2) &= \langle 0.7, 0.3 \rangle P(r_{2+k-1}|U_1, U_2) + \langle 0.3, 0.7 \rangle P(\neg r_{2+k-1}|U_1, U_2) \\
\mathbf{P}(r_{2+k}|U_1, U_2) &= 0.7 P(r_{2+k-1}|U_1, U_2) + 0.3(1 - P(r_{2+k-1}|U_1, U_2)) \\
&= 0.4 P(r_{2+k-1}|U_1, U_2) + 0.3
\end{aligned}$$

That is, $P(r_{2+k}|U_1, U_2) = 0.5$.

Notice that the fixed point does not depend on the initial evidence.

- 15.13 (Kushank) A professor wants to know if students are getting enough sleep. Each day the professor observes whether the students sleep in class, and whether they have red eyes. The professor has the following theory:

  - The prior probability of getting enough sleep, with no observations, is 0.7.

  - The probability of getting enough sleep on night t is 0.8 given that the student got enough sleep the previous night, and 0.3 if not.

  - The probability of having red eyes is 0.2 if the student got enough sleep, and 0.7 if not.

  - The probability of sleeping in class is 0.1 if the student got enough sleep, and 0.3 if not.

- Formulate this information as a dynamic Bayesian network that the professor could use to filter or predict from a sequence of observations. Give the complete probability tables for the model.

15.13 The DBN has three variables: S_t, whether the student gets enough sleep; R_t, whether they have red eyes in class; C_t, whether the student sleeps in class. S_t is a parent of S_{t+1}, R_t,

and C_t. The CPTs are given by

$$
\begin{aligned}
P(s_0) &= 0.7 \\
P(s_{t+1}|s_t) &= 0.8 \\
P(s_{t+1}|\neg s_t) &= 0.3 \\
P(r_t|s_t) &= 0.2 \\
P(r_t|\neg s_t) &= 0.7 \\
P(c_t|s_t) &= 0.1 \\
P(c_t|\neg s_t) &= 0.3
\end{aligned}
$$

To reformulate as an HMM with a single observation node, simply combine the 2-valued variables "having red eyes" and "sleeping in class" into a single 4-valued variable, multiplying together the emission probabilities. (Probability tables omitted.)

- (Kefan) For the 8-puzzle, define the state space, goal test and successor function. Write pseudocode to implement depth-first, breadth-first and A* search (using a reasonable heuristic).

- (Chun (Karen) Li) Consider a variant of tic-tac-toe where, if no one has won after 6 moves, the game is a draw. Draw a tic-tac-toe board and play out the first four moves randomly. Use the minimax algorithm to determine perfect for the last two moves.

- (Scott) Convert the following sentences to CNF.
  - $A \Leftrightarrow (B \vee E)$
  - $E \Rightarrow D$
  - $C \wedge F \Rightarrow \neg B$
  - $E \Rightarrow B$
  - $B \Rightarrow F$
  - $B \Rightarrow C$
- Trace the execution of DPLL on the conjunction of these clauses.

7.20 The CNF representations are as follows:

S1: $(\neg A \lor B \lor E) \land (\neg B \lor A) \land (\neg E \lor A)$.
S2: $(\neg E \lor D)$.
S3: $(\neg C \lor \neg F \lor \neg B)$.
S4: $(\neg E \lor B)$.
S5: $(\neg B \lor F)$.
S6: $(\neg B \lor C)$.

We omit the DPLL trace, which is easy to obtain from the version in the code repository.

- 8.23 (Nolan) For each of the following sentences in English, decide if the FOL sentence is a good translation. If not, explain why not and correct it. (Some sentences may have more than one error!)
  - No two people have the same Social Security Number.
    - $\neg\exists x,y,n\ Person(x) \wedge Person(y) \Rightarrow [HasSSN(x,n) \wedge HasSSN(y,n)]$
  - John's SSN is the same as Mary's.
    - $\exists n\ HasSSN(John, n) \wedge HasSSN(Mary,n)$
  - Everyone's SSN has nine digits.
    - $\forall x,n\ Person(x) \Rightarrow [HasSSN(x,n) \wedge Digits(n,9)]$
  - Rewrite each sentence above using a function *SSN(x)* instead of the predicate *HasSSN(x,n)*

8.23

a. "No two people have the same social security number."

$$\neg\exists\, x, y, n \;\; Person(x) \land Person(y) \;\Rightarrow\; [HasSS\#(x,n) \land HasSS\#(y,n)].$$

This uses $\Rightarrow$ with $\exists$. It also says that no person has a social security number because it doesn't restrict itself to the cases where x and y are not equal. Correct version:

$$\neg\exists\, x, y, n \;\; Person(x) \land Person(y) \land \neg(x = y) \land [HasSS\#(x,n) \land HasSS\#(y,n)]$$

b. "John's social security number is the same as Mary's."

$$\exists\, n \;\; HasSS\#(John, n) \land HasSS\#(Mary, n).$$

This is OK.

c. "Everyone's social security number has nine digits."

$$\forall\, x, n \;\; Person(x) \;\Rightarrow\; [HasSS\#(x,n) \land Digits(n, 9)].$$

This says that everyone has every number. $HasSS\#(x,n)$ should be in the premise:

$$\forall\, x, n \;\; Person(x) \land HasSS\#(x,n) \;\Rightarrow\; Digits(n, 9)$$

d. Here $SS\#(x)$ denotes the social security number of x. Using a function enforces the rule that everyone has just one.

$$\neg\exists\, x, y \;\; Person(x) \land Person(y) \;\Rightarrow\; [SS\#(x) = SS\#(y)]$$
$$SS\#(John) = SS\#(Mary)$$
$$\forall\, x \;\; Person(x) \;\Rightarrow\; Digits(SS\#(x), 9)$$

- (Alexander) Perform variable elimination on the Burglary network for the query P(Burglary | JohnCalls = true, MaryCalls = true).