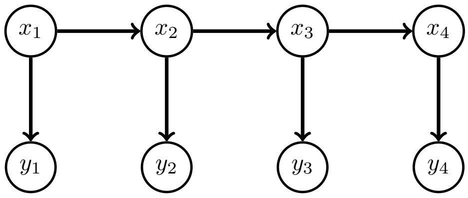
These are exercises for self-study and exam preparation. All material is examinable unless otherwise mentioned.

Exercise 1. *Conversion to factor graphs*

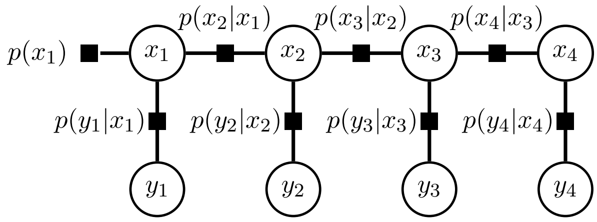
(a) *Draw an undirected factor graph for the directed graphical model defined by the graph below.*



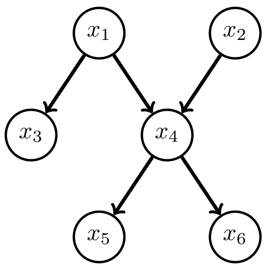
Solution. The graph specifies probabilistic models that factorise as

$$p(x_1, \dots, x_4, y_1, \dots, y_4) = p(x_1)p(y_1|x_1) \prod_{i=2}^{4} p(y_i|x_i)p(x_i|x_{i-1})$$

It is the graph for a hidden Markov model. The corresponding factor graph is shown below.



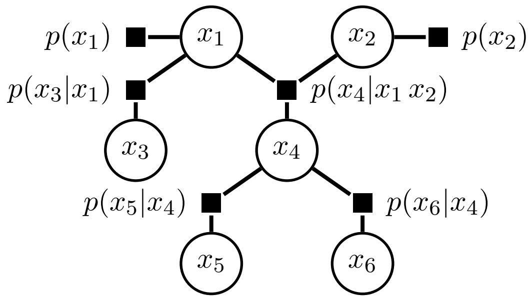
(b) *Draw an undirected factor graph for directed graphical models defined by the graph below (this kind of graph is called a polytree: there are no loops but a node may have more than one parent).*

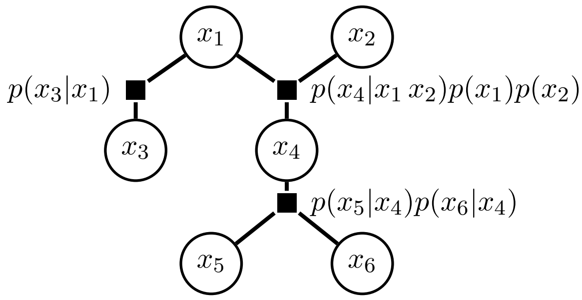


Solution. For the factor graph, we note that the directed graph defines the following class of probabilistic models

$$p(x_1, \dots x_6) = p(x_1)p(x_2)p(x_3|x_1)p(x_4|x_1, x_2)p(x_5|x_4)p(x_6|x_4)$$

This gives the factor graph on left in the figure below.



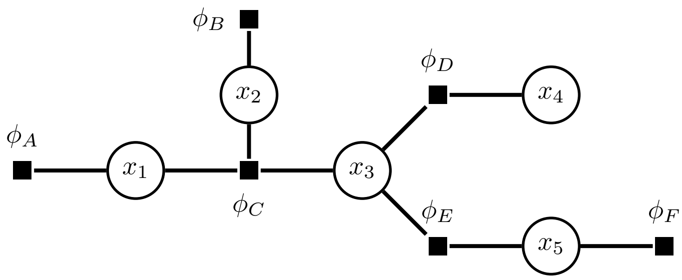


Note:

- One may drop the labels for the factors if the meaning is clear.
- One may choose to group some factors together in order to obtain a factor graph with a particular structure (see factor graph on right)

Exercise 2. ***Sum-product message passing***

We here re-consider the factor tree from the lecture on exact inference.



Let all variables be binary, $x_i \in \{0, 1\}$, and the factors be defined as follows:

x_1	ϕ_A
0	2
1	4

x_2	ϕ_B
0	4
1	4

x_1	x_2	x_3	ϕ_C
0	0	0	4
1	0	0	2
0	1	0	2
1	1	0	6
0	0	1	2
1	0	1	6
0	1	1	6
1	1	1	4

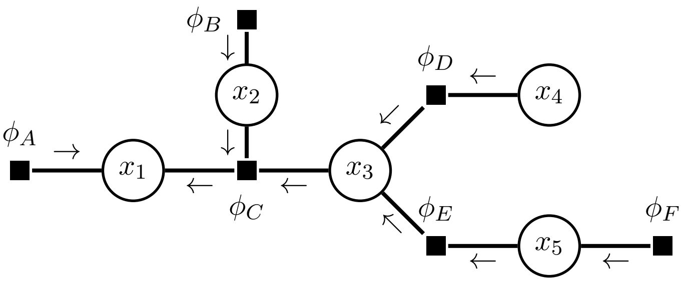
x_3	x_4	ϕ_D
0	0	8
1	0	2
0	1	2
1	1	6

x_3	x_5	ϕ_E
0	0	3
1	0	6
0	1	6
1	1	3

x_5	ϕ_F
0	1
1	8

(a) *Mark the graph with arrows indicating all messages that need to be computed for the computation of $p(x_1)$.*

Solution.



(b) Compute the messages that you have identified.

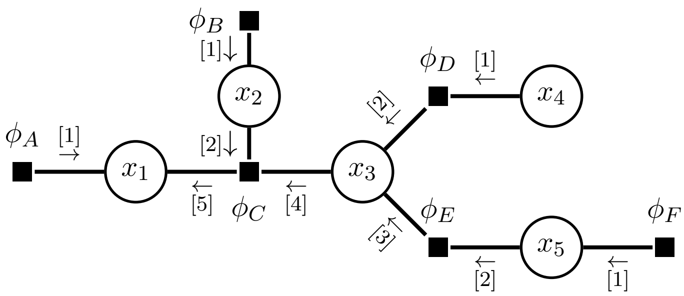
Assuming that the computation of the messages is scheduled according to a common clock, group the messages together so that all messages in the same group can be computed in parallel during a clock cycle.

Solution. Since the variables are binary, each message can be represented as a two-dimensional vector. We use the convention that the first element of the vector corresponds to the message for $x_i = 0$ and the second element to the message for $x_i = 1$. For example,

$$\boldsymbol{\mu}_{\boldsymbol{\phi_A \to x_1}} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} \tag{S.1}$$

means that the message $\mu_{\phi_A \to x_1}(x_1)$ equals 2 for $x_1 = 0$, i.e. $\mu_{\phi_A \to x_1}(0) = 2$.

The following figure shows a grouping (scheduling) of the computation of the messages.



Clock cycle 1:

$$\boldsymbol{\mu}_{\boldsymbol{\phi_A \to x_1}} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} \qquad \boldsymbol{\mu}_{\boldsymbol{\phi_B \to x_2}} = \begin{pmatrix} 4 \\ 4 \end{pmatrix} \qquad \boldsymbol{\mu}_{\boldsymbol{x_4 \to \phi_D}} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \boldsymbol{\mu}_{\boldsymbol{\phi_F \to x_5}} = \begin{pmatrix} 1 \\ 8 \end{pmatrix} \tag{S.2}$$

Clock cycle 2:

$$\boldsymbol{\mu}_{\boldsymbol{x_2 \to \phi_C}} = \boldsymbol{\mu}_{\boldsymbol{\phi_B \to x_2}} = \begin{pmatrix} 4 \\ 4 \end{pmatrix} \qquad \boldsymbol{\mu}_{\boldsymbol{x_5 \to \phi_E}} = \boldsymbol{\mu}_{\boldsymbol{\phi_F \to x_5}} = \begin{pmatrix} 1 \\ 8 \end{pmatrix} \tag{S.3}$$

Message $\mu_{\phi_D \to x_3}$ is defined as

$$\mu_{\phi_D \to x_3}(x_3) = \sum_{x_4} \phi_D(x_3, x_4)\mu_{x_4 \to \phi_D}(x_4) \tag{S.4}$$

so that

$$\begin{aligned}
\mu_{\phi_D\to x_3}(0) &= \sum_{x_4=0}^{1} \phi_D(0,x_4)\mu_{x_4\to\phi_D}(x_4) && \text{(S.5)}\\
&= \phi_D(0,0)\mu_{x_4\to\phi_D}(0) + \phi_D(0,1)\mu_{x_4\to\phi_D}(1) && \text{(S.6)}\\
&= 8\cdot 1 + 2\cdot 1 && \text{(S.7)}\\
&= 10 && \text{(S.8)}\\
\mu_{\phi_D\to x_3}(1) &= \sum_{x_4=0}^{1} \phi_D(1,x_4)\mu_{x_4\to\phi_D}(x_4) && \text{(S.9)}\\
&= \phi_D(1,0)\mu_{x_4\to\phi_D}(0) + \phi_D(1,1)\mu_{x_4\to\phi_D}(1) && \text{(S.10)}\\
&= 2\cdot 1 + 6\cdot 1 && \text{(S.11)}\\
&= 8 && \text{(S.12)}
\end{aligned}$$

and thus

$$\boldsymbol{\mu_{\phi_D\to x_3}} = \begin{pmatrix}10\\8\end{pmatrix}. \qquad \text{(S.13)}$$

The above computations can be written more compactly in matrix notation. Let $\boldsymbol{\phi_D}$ be the matrix that contains the outputs of $\phi_D(x_3, x_4)$

$$\boldsymbol{\phi_D} = \begin{pmatrix}\phi_D(x_3=0,x_4=0) & \phi_D(x_3=0,x_4=1)\\ \phi_D(x_3=1,x_4=0) & \phi_D(x_3=1,x_4=1)\end{pmatrix} = \begin{pmatrix}8 & 2\\ 2 & 6\end{pmatrix}. \qquad \text{(S.14)}$$

We can then write $\boldsymbol{\mu_{\phi_D\to x_3}}$ in terms of a matrix vector product,

$$\boldsymbol{\mu_{\phi_D\to x_3}} = \boldsymbol{\phi_D}\boldsymbol{\mu_{x_4\to\phi_D}}. \qquad \text{(S.15)}$$

Clock cycle 3:
Representing the factor ϕ_E as matrix $\boldsymbol{\phi_E}$,

$$\boldsymbol{\phi_E} = \begin{pmatrix}\phi_E(x_3=0,x_5=0) & \phi_E(x_3=0,x_5=1)\\ \phi_E(x_3=1,x_5=0) & \phi_E(x_3=1,x_5=1)\end{pmatrix} = \begin{pmatrix}3 & 6\\ 6 & 3\end{pmatrix}, \qquad \text{(S.16)}$$

we can write

$$\mu_{\phi_E\to x_3}(x_3) = \sum_{x_5} \phi_E(x_3,x_5)\mu_{x_5\to\phi_E}(x_5) \qquad \text{(S.17)}$$

as a matrix vector product,

$$\begin{aligned}
\boldsymbol{\mu_{\phi_E\to x_3}} &= \boldsymbol{\phi_E}\boldsymbol{\mu_{x_5\to\phi_E}} && \text{(S.18)}\\
&= \begin{pmatrix}3 & 6\\ 6 & 3\end{pmatrix}\begin{pmatrix}1\\8\end{pmatrix} && \text{(S.19)}\\
&= \begin{pmatrix}51\\30\end{pmatrix}. && \text{(S.20)}
\end{aligned}$$

Clock cycle 4:
Variable node x_3 has received all incoming messages, and can thus output $\mu_{x_3\to\phi_C}$,

$$\mu_{x_3\to\phi_C}(x_3) = \mu_{\phi_D\to x_3}(x_3)\mu_{\phi_E\to x_3}(x_3). \qquad \text{(S.21)}$$

Using $\odot$ to denote element-wise multiplication of two vectors, we have

$$\boldsymbol{\mu}_{x_3 \to \phi_C} = \boldsymbol{\mu}_{\phi_D \to x_3} \odot \boldsymbol{\mu}_{\phi_E \to x_3} \tag{S.22}$$

$$= \begin{pmatrix} 10 \\ 8 \end{pmatrix} \odot \begin{pmatrix} 51 \\ 30 \end{pmatrix} \tag{S.23}$$

$$= \begin{pmatrix} 510 \\ 240 \end{pmatrix}. \tag{S.24}$$

Clock cycle 5:
Factor node ϕ_C has received all incoming messages, and can thus output $\mu_{\phi_C \to x_1}$,

$$\mu_{\phi_C \to x_1}(x_1) = \sum_{x_2, x_3} \phi_C(x_1, x_2, x_3) \mu_{x_2 \to \phi_C}(x_2) \mu_{x_3 \to \phi_C}(x_3). \tag{S.25}$$

Writing out the sum for $x_1 = 0$ and $x_1 = 1$ gives

$$\mu_{\phi_C \to x_1}(0) = \sum_{x_2, x_3} \phi_C(0, x_2, x_3) \mu_{x_2 \to \phi_C}(x_2) \mu_{x_3 \to \phi_C}(x_3) \tag{S.26}$$

$$= \phi_C(0, x_2, x_3) \mu_{x_2 \to \phi_C}(x_2) \mu_{x_3 \to \phi_C}(x_3) \mid_{(x_2, x_3) = (0,0)} + \tag{S.27}$$

$$\phi_C(0, x_2, x_3) \mu_{x_2 \to \phi_C}(x_2) \mu_{x_3 \to \phi_C}(x_3) \mid_{(x_2, x_3) = (1,0)} + \tag{S.28}$$

$$\phi_C(0, x_2, x_3) \mu_{x_2 \to \phi_C}(x_2) \mu_{x_3 \to \phi_C}(x_3) \mid_{(x_2, x_3) = (0,1)} + \tag{S.29}$$

$$\phi_C(0, x_2, x_3) \mu_{x_2 \to \phi_C}(x_2) \mu_{x_3 \to \phi_C}(x_3) \mid_{(x_2, x_3) = (1,1)} \tag{S.30}$$

$$= 4 \cdot 4 \cdot 510 + \tag{S.31}$$

$$2 \cdot 4 \cdot 510 + \tag{S.32}$$

$$2 \cdot 4 \cdot 240 + \tag{S.33}$$

$$6 \cdot 4 \cdot 240 \tag{S.34}$$

$$= 19920 \tag{S.35}$$

$$\mu_{\phi_C \to x_1}(1) = \sum_{x_2, x_3} \phi_C(1, x_2, x_3) \mu_{x_2 \to \phi_C}(x_2) \mu_{x_3 \to \phi_C}(x_3) \tag{S.36}$$

$$= \phi_C(1, x_2, x_3) \mu_{x_2 \to \phi_C}(x_2) \mu_{x_3 \to \phi_C}(x_3) \mid_{(x_2, x_3) = (0,0)} + \tag{S.37}$$

$$\phi_C(1, x_2, x_3) \mu_{x_2 \to \phi_C}(x_2) \mu_{x_3 \to \phi_C}(x_3) \mid_{(x_2, x_3) = (1,0)} + \tag{S.38}$$

$$\phi_C(1, x_2, x_3) \mu_{x_2 \to \phi_C}(x_2) \mu_{x_3 \to \phi_C}(x_3) \mid_{(x_2, x_3) = (0,1)} + \tag{S.39}$$

$$\phi_C(1, x_2, x_3) \mu_{x_2 \to \phi_C}(x_2) \mu_{x_3 \to \phi_C}(x_3) \mid_{(x_2, x_3) = (1,1)} \tag{S.40}$$

$$= 2 \cdot 4 \cdot 510 + \tag{S.41}$$

$$6 \cdot 4 \cdot 510 + \tag{S.42}$$

$$6 \cdot 4 \cdot 240 + \tag{S.43}$$

$$4 \cdot 4 \cdot 240 \tag{S.44}$$

$$= 25920 \tag{S.45}$$

and hence

$$\boldsymbol{\mu}_{\phi_C \to x_1} = \begin{pmatrix} 19920 \\ 25920 \end{pmatrix} \tag{S.46}$$

After step 5, variable node x_1 has received all incoming messages and the marginal can be computed.

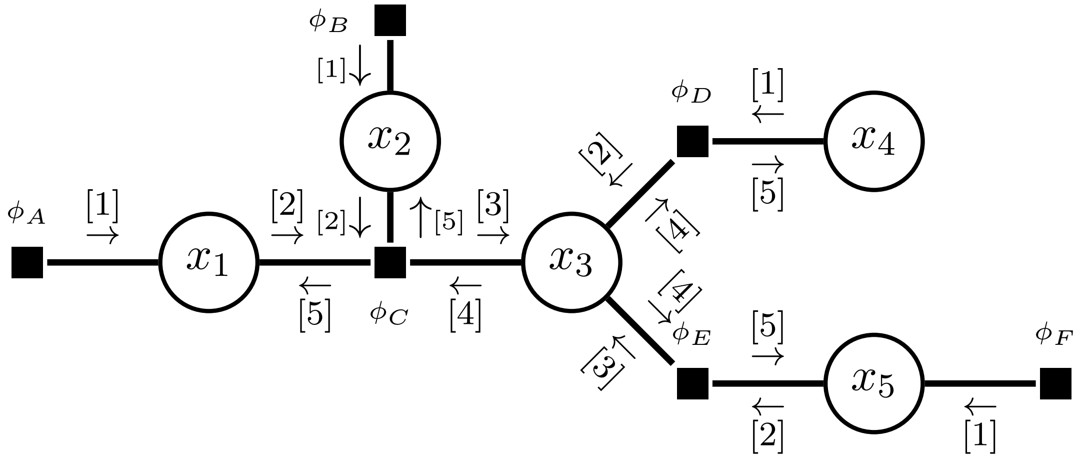


Figure 1: Answer to Exercise 2 Question (b): Computing all messages in five clock cycles. If we also computed the messages toward the leaf factor nodes, we needed six cycles, but they are not necessary for computation of the marginals so they are omitted.

In addition to the messages needed for computation of $p(x_1)$ one can compute *all* messages in the graph in five clock cycles, see Figure 1. This means that *all* marginals, as well as the joints of those variables sharing a factor node, are available after five clock cycles.

(c) *What is* $p(x_1 = 1)$*?*

Solution. We compute the marginal $p(x_1)$ as

$$p(x_1) \propto \mu_{\phi_A \to x_1}(x_1)\mu_{\phi_C \to x_1}(x_1) \tag{S.47}$$

which is in vector notation

$$\begin{pmatrix} p(x_1 = 0) \\ p(x_1 = 1) \end{pmatrix} \propto \boldsymbol{\mu_{\phi_A \to x_1}} \odot \boldsymbol{\mu_{\phi_C \to x_1}} \tag{S.48}$$

$$\propto \begin{pmatrix} 2 \\ 4 \end{pmatrix} \odot \begin{pmatrix} 19920 \\ 25920 \end{pmatrix} \tag{S.49}$$

$$\propto \begin{pmatrix} 39840 \\ 103680 \end{pmatrix}. \tag{S.50}$$

Normalisation gives

$$\begin{pmatrix} p(x_1 = 0) \\ p(x_1 = 1) \end{pmatrix} = \frac{1}{39840 + 103680} \begin{pmatrix} 39840 \\ 103680 \end{pmatrix} \tag{S.51}$$

$$= \begin{pmatrix} 0.2776 \\ 0.7224 \end{pmatrix} \tag{S.52}$$

so that $p(x_1 = 1) = 0.7224$.

Note the relatively large numbers in the messages that we computed. In other cases, one may obtain very small ones depending on the scale of the factors. This can cause numerical issues that can be addressed by working in the logarithmic domain.

(d) *Draw the factor graph corresponding to* $p(x_1, x_3, x_4, x_5 | x_2 = 1)$ *and provide the numerical values for all factors.*

Solution. The pmf represented by the original factor graph is

$$p(x_1, \dots, x_5) \propto \phi_A(x_1)\phi_B(x_2)\phi_C(x_1, x_2, x_3)\phi_D(x_3, x_4)\phi_E(x_3, x_5)\phi_F(x_5)$$

The conditional $p(x_1, x_3, x_4, x_5|x_2 = 1)$ is proportional to $p(x_1, \dots, x_5)$ with x_2 fixed to $x_2 = 1$, i.e.

$$p(x_1, x_3, x_4, x_5|x_2 = 1) \propto p(x_1, x_2 = 1, x_3, x_4, x_5) \tag{S.53}$$

$$\propto \phi_A(x_1)\phi_B(x_2 = 1)\phi_C(x_1, x_2 = 1, x_3)\phi_D(x_3, x_4)\phi_E(x_3, x_5)\phi_F(x_5) \tag{S.54}$$

$$\propto \phi_A(x_1)\phi_C^{x_2}(x_1, x_3)\phi_D(x_3, x_4)\phi_E(x_3, x_5)\phi_F(x_5) \tag{S.55}$$

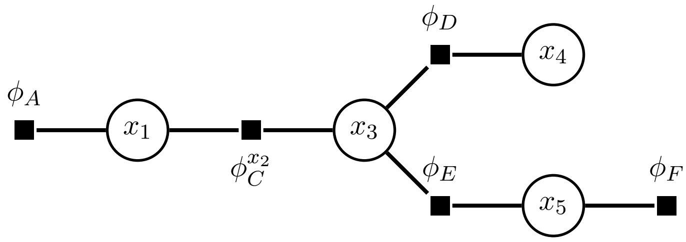
where $\phi_C^{x_2}(x_1, x_3) = \phi_C(x_1, x_2 = 1, x_3)$. The numerical values of $\phi_C^{x_2}(x_1, x_3)$ can be read from the table defining $\phi_C(x_1, x_2, x_3)$, extracting those rows where $x_2 = 1$,

	x_1	x_2	x_3	ϕ_C
	0	0	0	4
	1	0	0	2
→	0	1	0	2
→	1	1	0	6
	0	0	1	2
	1	0	1	6
→	0	1	1	6
→	1	1	1	4

so that

x_1	x_3	$\phi_C^{x_2}$
0	0	2
1	0	6
0	1	6
1	1	4

The factor graph for $p(x_1, x_3, x_4, x_5|x_2 = 1)$ is shown below. Factor ϕ_B has disappeared since it only depended on x_2 and thus became a constant. Factor ϕ_C is replaced by $\phi_C^{x_2}$ defined above. The remaining factors are the same as in the original factor graph.



(e) *Compute $p(x_1 = 1|x_2 = 1)$, re-using messages that you have already computed for the evaluation of $p(x_1 = 1)$.*

Solution. The message $\mu_{\phi_A \to x_1}$ is the same as in the original factor graph and $\mu_{x_3 \to \phi_C^{x_2}} = \mu_{x_3 \to \phi_C}$. This is because the outgoing message from x_3 corresponds to the effective factor obtained by summing out all variables in the sub-trees attached to x_3 (without the $\phi_C^{x_2}$ branch), and these sub-trees do not depend on x_2.

The message $\mu_{\phi_C^{x_2} \to x_1}$ needs to be newly computed. We have

$$\mu_{\phi_C^{x_2} \to x_1}(x_1) = \sum_{x_3} \phi_C^{x_2}(x_1, x_3)\mu_{x_3 \to \phi_C^{x_2}} \tag{S.56}$$

or in vector notation

$$\boldsymbol{\mu}_{\phi_C^{x_2} \to x_1} = \boldsymbol{\phi}_C^{x_2} \boldsymbol{\mu}_{x_3 \to \phi_C^{x_2}} \tag{S.57}$$

$$= \begin{pmatrix} \phi_C^{x_2}(x_1 = 0, x_3 = 0) & \phi_C^{x_2}(x_1 = 0, x_3 = 1) \\ \phi_C^{x_2}(x_1 = 1, x_3 = 0) & \phi_C^{x_2}(x_1 = 1, x_3 = 1) \end{pmatrix} \boldsymbol{\mu}_{x_3 \to \phi_C^{x_2}} \tag{S.58}$$

$$= \begin{pmatrix} 2 & 6 \\ 6 & 4 \end{pmatrix} \begin{pmatrix} 510 \\ 240 \end{pmatrix} \tag{S.59}$$

$$= \begin{pmatrix} 2460 \\ 4020 \end{pmatrix} \tag{S.60}$$

We thus obtain for the marginal posterior of x_1 given $x_2 = 1$:

$$\begin{pmatrix} p(x_1 = 0|x_2 = 1) \\ p(x_1 = 1|x_2 = 1) \end{pmatrix} \propto \boldsymbol{\mu}_{\phi_A \to x_1} \odot \boldsymbol{\mu}_{\phi_C^{x_2} \to x_1} \tag{S.61}$$

$$\propto \begin{pmatrix} 2 \\ 4 \end{pmatrix} \odot \begin{pmatrix} 2460 \\ 4020 \end{pmatrix} \tag{S.62}$$

$$\propto \begin{pmatrix} 4920 \\ 16080 \end{pmatrix}. \tag{S.63}$$

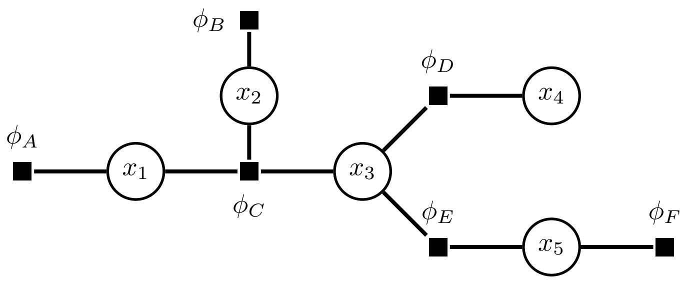
Normalisation gives

$$\begin{pmatrix} p(x_1 = 0|x_2 = 1) \\ p(x_1 = 1|x_2 = 1) \end{pmatrix} = \begin{pmatrix} 0.2343 \\ 0.7657 \end{pmatrix} \tag{S.64}$$

and thus $p(x_1 = 1|x_2 = 1) = 0.7657$. The posterior probability is slightly larger than the prior probability, $p(x_1 = 1) = 0.7224$.

Exercise 3. *Max-sum message passing*

We here compute most probable states for the factor graph and factors below.



Let all variables be binary, $x_i \in \{0, 1\}$, and the factors be defined as follows:

x_1	ϕ_A
0	2
1	4

x_2	ϕ_B
0	4
1	4

x_1	x_2	x_3	ϕ_C
0	0	0	4
1	0	0	2
0	1	0	2
1	1	0	6
0	0	1	2
1	0	1	6
0	1	1	6
1	1	1	4

x_3	x_4	ϕ_D
0	0	8
1	0	2
0	1	2
1	1	6

x_3	x_5	ϕ_E
0	0	3
1	0	6
0	1	6
1	1	3

x_5	ϕ_F
0	1
1	8

(a) *Will we need to compute the normalising constant Z to determine* $\operatorname{argmax}_{\mathbf{x}} p(x_1, \ldots, x_5)$*?*

Solution. This is not necessary since $\operatorname{argmax}_{\mathbf{x}} p(x_1, \ldots, x_5) = \operatorname{argmax}_{\mathbf{x}} cp(x_1, \ldots, x_5)$ for any constant c. Algorithmically, the backtracking algorithm is also invariant to any scaling of the factors.

(b) *Compute* $\operatorname{argmax}_{x_1,x_2,x_3} p(x_1, x_2, x_3|x_4 = 0, x_5 = 0)$ *via max-sum message passing.*

Solution. We first derive the factor graph and corresponding factors for $p(x_1, x_2, x_3|x_4 = 0, x_5 = 0)$.

For fixed values of x_4, x_5, the two variables are removed from the graph, and the factors $\phi_D(x_3, x_4)$ and $\phi_E(x_3, x_5)$ are reduced to univariate factors $\phi_D^{x_4}(x_3)$ and $\phi_D^{x_5}(x_3)$ by retaining those rows in the table where $x_4 = 0$ and $x_5 = 0$, respectively:

x_3	$\phi_D^{x_4}$
0	8
1	2

x_3	$\phi_E^{x_5}$
0	3
1	6

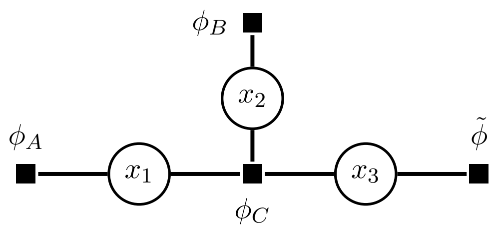
Since both factors only depend on x_3, they can be combined into a new factor $\tilde{\phi}(x_3)$ by element-wise multiplication.

x_3	$\tilde{\phi}$
0	24
1	12

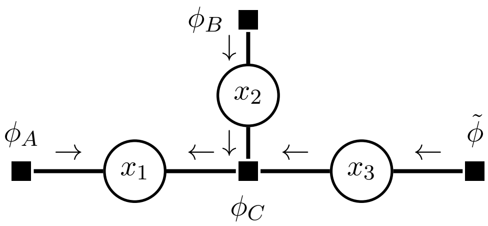
Moreover, since we work with an unnormalised model, we can rescale the factor so that the maximum value is one, so that

x_3	$\tilde{\phi}$
0	2
1	1

Factor $\phi_F(x_5)$ is a constant for fixed value of x_5 and can be ignored. The factor graph for $p(x_1, x_2, x_3|x_4 = 0, x_5 = 0)$ thus is



Let us fix x_1 as sink node towards which we compute the messages. The messages that we need to compute are shown in the following graph



Next, we compute the leaf (log) messages. We only have factor nodes as leaf nodes so that

$$\boldsymbol{\lambda}_{\phi_A \to x_1} = \begin{pmatrix} \log \phi_A(x_1 = 0) \\ \log \phi_A(x_1 = 1) \end{pmatrix} = \begin{pmatrix} \log 2 \\ \log 4 \end{pmatrix} \tag{S.65}$$

and similarly

$$\boldsymbol{\lambda}_{\phi_B \to x_2} = \begin{pmatrix} \log \phi_B(x_2 = 0) \\ \log \phi_B(x_2 = 1) \end{pmatrix} = \begin{pmatrix} \log 4 \\ \log 4 \end{pmatrix} \quad \boldsymbol{\lambda}_{\tilde{\phi} \to x_3} = \begin{pmatrix} \log \tilde{\phi}(x_3 = 0) \\ \log \tilde{\phi}(x_3 = 1) \end{pmatrix} = \begin{pmatrix} \log 2 \\ \log 1 \end{pmatrix} \tag{S.66}$$

Since the variable nodes x_2 and x_3 only have one incoming edge each, we obtain

$$\boldsymbol{\lambda}_{x_2 \to \phi_C} = \boldsymbol{\lambda}_{\phi_B \to x_2} = \begin{pmatrix} \log 4 \\ \log 4 \end{pmatrix} \qquad \boldsymbol{\lambda}_{x_3 \to \phi_C} = \boldsymbol{\lambda}_{\tilde{\phi} \to x_3} = \begin{pmatrix} \log 2 \\ \log 1 \end{pmatrix} \tag{S.67}$$

The message $\lambda_{\phi_C \to x_1}(x_1)$ equals

$$\lambda_{\phi_C \to x_1}(x_1) = \max_{x_2, x_3} \log \phi_C(x_1, x_2, x_3) + \lambda_{x_2 \to \phi_C}(x_2) + \lambda_{x_3 \to \phi_C}(x_3) \tag{S.68}$$

where we wrote the messages in non-vector notation to highlight their dependency on the variables x_2 and x_3. We now have to consider all combinations of x_2 and x_3

x_2	x_3	$\log \phi_C(x_1 = 0, x_2, x_3)$
0	0	$\log 4$
1	0	$\log 2$
0	1	$\log 2$
1	1	$\log 6$

x_2	x_3	$\log \phi_C(x_1 = 1, x_2, x_3)$
0	0	$\log 2$
1	0	$\log 6$
0	1	$\log 6$
1	1	$\log 4$

Furthermore

x_2	x_3	$\lambda_{x_2 \to \phi_C}(x_2) + \lambda_{x_3 \to \phi_C}(x_3)$
0	0	$\log 4 + \log 2 = \log 8$
1	0	$\log 4 + \log 2 = \log 8$
0	1	$\log 4$
1	1	$\log 4$

Hence for $x_1 = 0$, we have

x_2	x_3	$\log \phi_C(x_1 = 0, x_2, x_3) + \lambda_{x_2 \to \phi_C}(x_2) + \lambda_{x_3 \to \phi_C}(x_3)$
0	0	$\log 4 + \log 8 = \log 32$
1	0	$\log 2 + \log 8 = \log 16$
0	1	$\log 2 + \log 4 = \log 8$
1	1	$\log 6 + \log 4 = \log 24$

The maximal value is $\log 32$ and for backtracking, we also need to keep track of the argmax which is here $\hat{x}_2 = \hat{x}_3 = 0$.

For $x_1 = 1$, we have

x_2	x_3	$\log \phi_C(x_1 = 1, x_2, x_3) + \lambda_{x_2 \to \phi_C}(x_2) + \lambda_{x_3 \to \phi_C}(x_3)$
0	0	$\log 2 + \log 8 = \log 16$
1	0	$\log 6 + \log 8 = \log 48$
0	1	$\log 6 + \log 4 = \log 24$
1	1	$\log 4 + \log 4 = \log 16$

The maximal value is $\log 48$ and the argmax is $(\hat{x}_2 = 1, \hat{x}_3 = 0)$.

So overall, we have

$$\boldsymbol{\lambda}_{\phi_C \to x_1} = \begin{pmatrix} \lambda_{\phi_C \to x_1}(x_1 = 0) \\ \lambda_{\phi_C \to x_1}(x_1 = 1) \end{pmatrix} = \begin{pmatrix} \log 32 \\ \log 48 \end{pmatrix} \tag{S.69}$$

and the argmax back-tracking function is

$$\lambda^*_{\phi_C \to x_1}(x_1) = \begin{cases} (\hat{x}_2 = 0, \hat{x}_3 = 0) & \text{if } x_1 = 0 \\ (\hat{x}_2 = 1, \hat{x}_3 = 0) & \text{if } x_1 = 1 \end{cases} \tag{S.70}$$

We now have all incoming messages to the assigned sink node x_1. *Ignoring the normalising constant*, we obtain

$$\boldsymbol{\gamma} = \begin{pmatrix} \gamma^*(x_1 = 0) \\ \gamma^*(x_1 = 1) \end{pmatrix} = \boldsymbol{\lambda}_{\phi_A \to x_1} + \boldsymbol{\lambda}_{\phi_C \to x_1} \tag{S.71}$$

$$= \begin{pmatrix} \log 2 \\ \log 4 \end{pmatrix} + \begin{pmatrix} \log 32 \\ \log 48 \end{pmatrix} = \begin{pmatrix} \log 64 \\ \log 192 \end{pmatrix} \tag{S.72}$$

The value x_1 for which $\gamma^*(x_1)$ is largest is thus $\hat{x}_1 = 1$. Plugging $\hat{x}_1 = 1$ into the back-tracking function $\lambda^*_{\phi_C \to x_1}(x_1)$ gives

$$(\hat{x}_1, \hat{x}_2, \hat{x}_3) = \underset{x_1, x_2, x_3}{\operatorname{argmax}}\, p(x_1, x_2, x_3 | x_4 = 0, x_5 = 0) = (1, 1, 0). \tag{S.73}$$

In this low-dimensional example, we can verify the solution by computing the unnormalised pmf for all combinations of x_1, x_2, x_3. This is done in the following table where we start with the table for ϕ_C and then multiply-in the further factors ϕ_A, $\tilde{\phi}$ and ϕ_B.

x_1	x_2	x_3	ϕ_C	$\phi_C \phi_A$	$\phi_C \phi_A \tilde{\phi}$	$\phi_C \phi_A \tilde{\phi} \phi_B$
0	0	0	4	8	16	$16 \cdot 4$
1	0	0	2	8	16	$16 \cdot 4$
0	1	0	2	4	8	$8 \cdot 4$
1	1	0	6	24	48	$48 \cdot 4$
0	0	1	2	4	4	$4 \cdot 4$
1	0	1	6	24	24	$24 \cdot 4$
0	1	1	6	12	12	$12 \cdot 4$
1	1	1	4	16	16	$16 \cdot 4$

For example, for the column $\phi_c \phi_A$, we multiply each value of $\phi_C(x_1, x_2, x_3)$ by $\phi_A(x_1)$, so that the rows with $x_1 = 0$ get multiplied by 2, and the rows with $x_1 = 1$ by 4.

The maximal value in the final column is achieved for $x_1 = 1, x_2 = 1, x_3 = 0$, in line with the result above (and $48 \cdot 4 = 192$). Since $\phi_B(x_2)$ is a constant, being equal to 4 for all values of x_2, we could have ignored it in the computation. The formal reason for this is that since the model is unnormalised, we are allowed to rescale each factor by an arbitrary (factor-dependent) *constant*. This operation does not change the model. So we could divide ϕ_B by 4 which would give a value of 1, so that the factor can indeed be ignored.

(c) *Compute* $\operatorname{argmax}_{x_1,\ldots,x_5} p(x_1, \ldots, x_5)$ *via max-sum message passing with* x_1 *as sink.*

Solution. As discussed in the solution to the answer above, we can drop factor $\phi_B(x_2)$ since it takes the same value for all x_2. Moreover, we can rescale the individual factors by a constant so they are more amenable to calculations by hand. We normalise them such that the largest value is one, which gives the following factors. Note that this is entirely optional.

x_1	ϕ_A
0	1
1	2

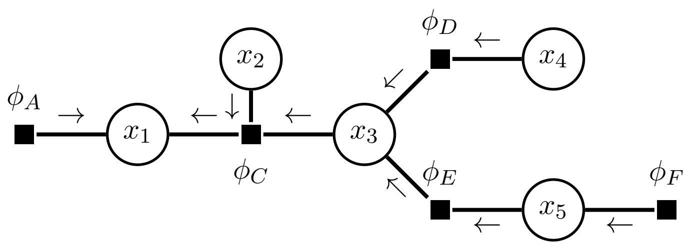
x_1	x_2	x_3	ϕ_C
0	0	0	2
1	0	0	1
0	1	0	1
1	1	0	3
0	0	1	1
1	0	1	3
0	1	1	3
1	1	1	2

x_3	x_4	ϕ_D
0	0	4
1	0	1
0	1	1
1	1	3

x_3	x_5	ϕ_E
0	0	1
1	0	2
0	1	2
1	1	1

x_5	ϕ_F
0	1
1	8

The factor graph without ϕ_B together with the messages that we need to compute is:



The leaf (log) messages are (using vector notation where the top element corresponds to $x_i = 0$ and the bottom one to $x_i = 1$):

$$\boldsymbol{\lambda}_{\phi_A \to x_1} = \begin{pmatrix} 0 \\ \log 2 \end{pmatrix} \quad \boldsymbol{\lambda}_{x_2 \to \phi_C} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \boldsymbol{\lambda}_{x_4 \to \phi_D} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \boldsymbol{\lambda}_{\phi_F \to x_5} = \begin{pmatrix} 0 \\ \log 8 \end{pmatrix} \tag{S.74}$$

The variable node x_5 only has one incoming edge so that $\boldsymbol{\lambda}_{x_5 \to \phi_E} = \boldsymbol{\lambda}_{\phi_F \to x_5}$. The message $\lambda_{\phi_E \to x_3}(x_3)$ equals

$$\lambda_{\phi_E \to x_3}(x_3) = \max_{x_5} \log \phi_E(x_3, x_5) + \lambda_{x_5 \to \phi_E}(x_5) \tag{S.75}$$

Writing out $\log \phi_E(x_3, x_5) + \lambda_{x_5 \to \phi_E}(x_5)$ for all x_5 as a function of x_3 we have

x_5	$\log \phi_E(x_3 = 0, x_5) + \lambda_{x_5 \to \phi_E}(x_5)$
0	$\log 1 + 0 = 0$
1	$\log 2 + \log 8 = \log 16$

x_5	$\log \phi_E(x_3 = 1, x_5) + \lambda_{x_5 \to \phi_E}(x_5)$
0	$\log 2 + 0 = \log 2$
1	$\log 1 + \log 8 = \log 8$

Taking the maximum over x_5 as a function of x_3, we obtain

$$\boldsymbol{\lambda}_{\phi_E \to x_3} = \begin{pmatrix} \log 16 \\ \log 8 \end{pmatrix} \tag{S.76}$$

and the backtracking function that indicates the maximiser $\hat{x}_5 = \text{argmax}_{x_5} \log \phi_E(x_3, x_5) + \lambda_{x_5 \to \phi_E}(x_5)$ as a function of x_3 equals

$$\lambda^*_{\phi_E \to x_3}(x_3) = \begin{cases} \hat{x}_5 = 1 & \text{if } x_3 = 0 \\ \hat{x}_5 = 1 & \text{if } x_3 = 1 \end{cases} \tag{S.77}$$

We perform the same kind of operation for $\lambda_{\phi_D \to x_3}(x_3)$

$$\lambda_{\phi_D \to x_3}(x_3) = \max_{x_4} \log \phi_D(x_3, x_4) + \lambda_{x_4 \to \phi_D}(x_4) \tag{S.78}$$

Since $\lambda_{x_4 \to \phi_D}(x_4) = 0$ for all x_4, the table with all values of $\log \phi_D(x_3, x_4) + \lambda_{x_4 \to \phi_D}(x_4)$ is

x_3	x_4	$\log \phi_D(x_3, x_4) + \lambda_{x_4 \to \phi_D}(x_4)$
0	0	$\log 4 + 0 = \log 4$
1	0	$\log 1 + 0 = 0$
0	1	$\log 1 + 0 = 0$
1	1	$\log 3 + 0 = \log 3$

Taking the maximum over x_4 as a function of x_3 we thus obtain

$$\boldsymbol{\lambda}_{\phi_D \to x_3} = \begin{pmatrix} \log 4 \\ \log 3 \end{pmatrix} \tag{S.79}$$

and the backtracking function that indicates the maximiser $\hat{x}_4 = \text{argmax}_{x_4} \log \phi_D(x_3, x_4) + \lambda_{x_4 \to \phi_D}(x_4)$ as a function of x_3 equals

$$\lambda^*_{\phi_D \to x_3}(x_3) = \begin{cases} \hat{x}_4 = 0 & \text{if } x_3 = 0 \\ \hat{x}_4 = 1 & \text{if } x_3 = 1 \end{cases} \tag{S.80}$$

For the message $\lambda_{x_3 \to \phi_C}(x_3)$ we add together the messages $\lambda_{\phi_E \to x_3}(x_3)$ and $\lambda_{\phi_D \to x_3}(x_3)$ which gives

$$\boldsymbol{\lambda}_{x_3 \to \phi_C} = \begin{pmatrix} \log 16 + \log 4 \\ \log 8 + \log 3 \end{pmatrix} = \begin{pmatrix} \log 64 \\ \log 24 \end{pmatrix} \tag{S.81}$$

Next we compute the message $\lambda_{\phi_C \to x_1}(x_1)$ by maximising over x_2 and x_3,

$$\lambda_{\phi_C \to x_1}(x_1) = \max_{x_2, x_3} \log \phi_C(x_1, x_2, x_3) + \lambda_{x_2 \to \phi_C}(x_2) + \lambda_{x_3 \to \phi_C}(x_3) \tag{S.82}$$

Since $\lambda_{x_2 \to \phi_C}(x_2) = 0$, the problem becomes

$$\lambda_{\phi_C \to x_1}(x_1) = \max_{x_2, x_3} \log \phi_C(x_1, x_2, x_3) + \lambda_{x_3 \to \phi_C}(x_3) \tag{S.83}$$

Building on the table for ϕ_C, we form a table with all values of $\log \phi_C(x_1, x_2, x_3) + \lambda_{x_3 \to \phi_C}(x_3)$

x_1	x_2	x_3	$\log \phi_C(x_1, x_2, x_3) + \lambda_{x_3 \to \phi_C}(x_3)$
0	0	0	$\log 2 + \log 64 = \mathbf{\log 128}$
1	0	0	$0 + \log 64 = \log 64$
0	1	0	$0 + \log 64 = \log 64$
1	1	0	$\log 3 + \log 64 = \mathbf{\log 192}$
0	0	1	$\log 24$
1	0	1	$\log 3 + \log 24 = \log 72$
0	1	1	$\log 3 + \log 24 = \log 72$
1	1	1	$\log 2 + \log 24 = \log 48$

The maximal value as a function of x_1 are highlighted in the table, which gives the message

$$\boldsymbol{\lambda}_{\phi_C \to x_1} = \begin{pmatrix} \log 128 \\ \log 192 \end{pmatrix} \tag{S.84}$$

and the backtracking function

$$\lambda^*_{\phi_C \to x_1}(x_1) = \begin{cases} (\hat{x}_2 = 0, \hat{x}_3 = 0) & \text{if } x_1 = 0 \\ (\hat{x}_2 = 1, \hat{x}_3 = 0) & \text{if } x_1 = 1 \end{cases} \tag{S.85}$$

We now have all incoming messages to the assigned sink node x_1. *Ignoring the normalising constant*, we obtain

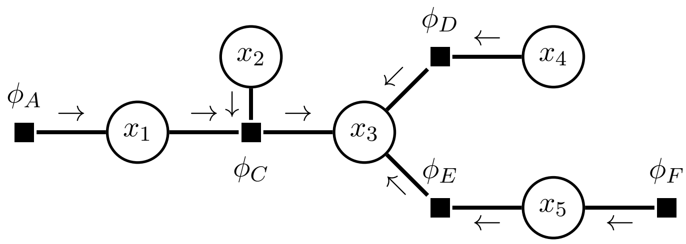
$$\boldsymbol{\gamma} = \begin{pmatrix} \gamma^*(x_1 = 0) \\ \gamma^*(x_1 = 1) \end{pmatrix} = \begin{pmatrix} 0 + \log 128 \\ \log 2 + \log 192 \end{pmatrix} \tag{S.86}$$

We can now start the backtracking to compute the desired $\text{argmax}_{x_1,\ldots,x_5} p(x_1, \ldots, x_5)$. Starting at the sink we have $\hat{x}_1 = \text{argmax}_{x_1} \gamma^*(x_1) = 1$. Plugging this value into the look-up table $\lambda^*_{\phi_C \to x_1}(x_1)$, we obtain $(\hat{x}_2 = 1, \hat{x}_3 = 0)$. With the look-up table $\lambda^*_{\phi_E \to x_3}(x_3)$ we find $\hat{x}_5 = 1$ and $\lambda^*_{\phi_D \to x_3}(x_3)$ gives $\hat{x}_4 = 0$ so that overall

$$\underset{x_1,\ldots,x_5}{\text{argmax}}\, p(x_1, \ldots, x_5) = (1, 1, 0, 0, 1). \tag{S.87}$$

(d) *Compute* $\text{argmax}_{x_1,\ldots,x_5} p(x_1, \ldots, x_5)$ *via max-sum message passing with* x_3 *as sink.*

Solution. With x_3 as sink, we need the following messages:



The following messages are the same as when x_1 was the sink:

$$\boldsymbol{\lambda}_{\phi_D \to x_3} = \begin{pmatrix} \log 4 \\ \log 3 \end{pmatrix} \quad \boldsymbol{\lambda}_{\phi_E \to x_3} = \begin{pmatrix} \log 16 \\ \log 8 \end{pmatrix} \quad \boldsymbol{\lambda}_{\phi_A \to x_1} = \begin{pmatrix} 0 \\ \log 2 \end{pmatrix} \quad \boldsymbol{\lambda}_{x_2 \to \phi_C} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{S.88}$$

Since x_1 has only one incoming message, we further have

$$\boldsymbol{\lambda}_{x_1 \to \phi_C} = \boldsymbol{\lambda}_{\phi_A \to x_1} = \begin{pmatrix} 0 \\ \log 2 \end{pmatrix}. \tag{S.89}$$

We next compute $\lambda_{\phi_C \to x_3}(x_3)$,

$$\lambda_{\phi_C \to x_3}(x_3) = \max_{x_1, x_2} \log \phi_C(x_1, x_2, x_3) + \lambda_{x_1 \to \phi_C}(x_1) + \lambda_{x_2 \to \phi_C}(x_2). \tag{S.90}$$

We first form a table for $\log \phi_C(x_1, x_2, x_3) + \lambda_{x_1 \to \phi_C}(x_1) + \lambda_{x_2 \to \phi_C}(x_2)$ noting that $\lambda_{x_2 \to \phi_C}(x_2) = 0$

x_1	x_2	x_3	$\log \phi_C(x_1, x_2, x_3) + \lambda_{x_1 \to \phi_C}(x_1) + \lambda_{x_2 \to \phi_C}(x_2)$
0	0	0	$\log 2 + 0 = \log 2$
1	0	0	$0 + \log 2 = \log 2$
0	1	0	$0 + 0 = 0$
1	1	0	$\log 3 + \log 2 = \mathbf{log\ 6}$
0	0	1	$0 + 0 = 0$
1	0	1	$\log 3 + \log 2 = \mathbf{log\ 6}$
0	1	1	$\log 3 + 0 = \log 3$
1	1	1	$\log 2 + \log 2 = \log 4$

The maximal value as a function of x_3 are highlighted in the table, which gives the message

$$\boldsymbol{\lambda}_{\phi_C \to x_3} = \begin{pmatrix} \log 6 \\ \log 6 \end{pmatrix} \tag{S.91}$$

and the backtracking function

$$\lambda^*_{\phi_C \to x_3}(x_3) = \begin{cases} (\hat{x}_1 = 1, \hat{x}_2 = 1) & \text{if } x_3 = 0 \\ (\hat{x}_1 = 1, \hat{x}_2 = 0) & \text{if } x_3 = 1 \end{cases} \tag{S.92}$$

We have now all incoming messages for x_3 and can compute $\gamma^*(x_3)$ up the normalising constant $-\log Z$ (which is not needed if we are interested in the argmax only:

$$\boldsymbol{\gamma} = \begin{pmatrix} \gamma^*(x_3 = 0) \\ \gamma^*(x_3 = 1) \end{pmatrix} = \boldsymbol{\lambda}_{\phi_C \to x_3} + \boldsymbol{\lambda}_{\phi_D \to x_3} + \boldsymbol{\lambda}_{\phi_E \to x_3} \tag{S.93}$$

$$= \begin{pmatrix} \log 6 + \log 4 + \log 16 = \log 384 \\ \log 6 + \log 3 + \log 8 = \log 144 \end{pmatrix} \tag{S.94}$$

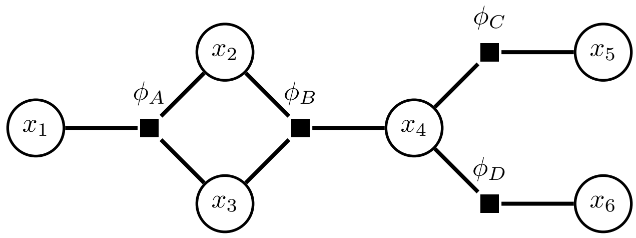
We can now start the backtracking which gives: $\hat{x}_3 = 0$, so that $\lambda^*_{\phi_C \to x_3}(0) = (\hat{x}_1 = 1, \hat{x}_2 = 1)$. The backtracking functions $\lambda^*_{\phi_E \to x_3}(x_3)$ and $\lambda^*_{\phi_D \to x_3}(x_3)$ are the same for question (c), which gives $\lambda^*_{\phi_E \to x_3}(0) = \hat{x}_5 = 1$ and $\lambda^*_{\phi_D \to x_3}(0) = \hat{x}_4 = 0$. Hence, overall, we find

$$\operatorname*{argmax}_{x_1, \ldots, x_5} p(x_1, \ldots, x_5) = (1, 1, 0, 0, 1). \tag{S.95}$$

Note that this matches the result from question (c) where x_1 was the sink. This is because the output of the max-sum algorithm is invariant to the choice of the sink.

Exercise 4. ***Choice of elimination order in factor graphs***

Consider the following factor graph, which contains a loop:



Let all variables be binary, $x_i \in \{0, 1\}$, and the factors be defined as follows:

x_1	x_2	x_3	ϕ_A
0	0	0	4
1	0	0	2
0	1	0	2
1	1	0	6
0	0	1	2
1	0	1	6
0	1	1	6
1	1	1	4

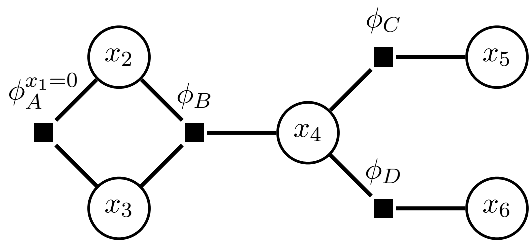
x_2	x_3	x_4	ϕ_B
0	0	0	2
1	0	0	2
0	1	0	4
1	1	0	2
0	0	1	6
1	0	1	8
0	1	1	4
1	1	1	2

x_4	x_5	ϕ_C
0	0	8
1	0	2
0	1	2
1	1	6

x_4	x_6	ϕ_D
0	0	3
1	0	6
0	1	6
1	1	3

(a) *Draw the factor graph corresponding to $p(x_2, x_3, x_4, x_5 \mid x_1 = 0, x_6 = 1)$ and give the tables defining the new factors $\phi_A^{x_1=0}(x_2, x_3)$ and $\phi_D^{x_6=1}(x_4)$ that you obtain.*

Solution. First condition on $x_1 = 0$:
Factor node $\phi_A(x_1, x_2, x_3)$ depends on x_1, thus we create a new factor $\phi_A^{x_1=0}(x_2, x_3)$ from the table for ϕ_A using the rows where $x_1 = 0$.



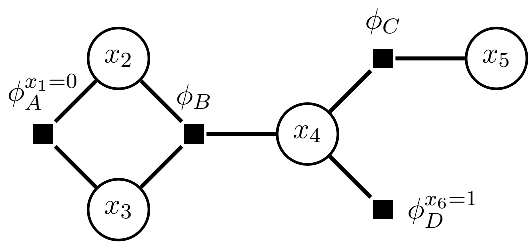
	x_1	x_2	x_3	ϕ_A
$\rightarrow$	0	0	0	4
	1	0	0	2
$\rightarrow$	0	1	0	2
	1	1	0	6
$\rightarrow$	0	0	1	2
	1	0	1	6
$\rightarrow$	0	1	1	6
	1	1	1	4

so that

x_2	x_3	$\phi_A^{x_1=0}$
0	0	4
1	0	2
0	1	2
1	1	6

Next condition on $x_6 = 1$:

Factor node $\phi_D(x_4, x_6)$ depends on x_6, thus we create a new factor $\phi_D^{x_6=1}(x_4)$ from the table for ϕ_D using the rows where $x_6 = 1$.



	x_4	x_6	ϕ_D
	0	0	3
	1	0	6
→	0	1	6
→	1	1	3

so that

x_4	$\phi_D^{x_6=1}$
0	6
1	3

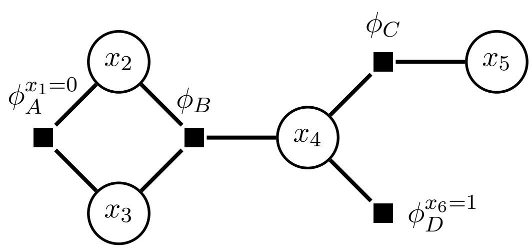
(b) *Find $p(x_2 \mid x_1 = 0, x_6 = 1)$ using the elimination ordering (x_4, x_5, x_3):*

(i) *Draw the graph for $p(x_2, x_3, x_5 \mid x_1 = 0, x_6 = 1)$ by marginalising x_4*
Compute the table for the new factor $\tilde{\phi}_4(x_2, x_3, x_5)$

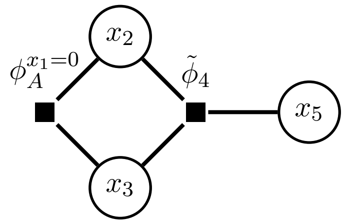
(ii) *Draw the graph for $p(x_2, x_3 \mid x_1 = 0, x_6 = 1)$ by marginalising x_5*
Compute the table for the new factor $\tilde{\phi}_{45}(x_2, x_3)$

(iii) *Draw the graph for $p(x_2 \mid x_1 = 0, x_6 = 1)$ by marginalising x_3*
Compute the table for the new factor $\tilde{\phi}_{453}(x_2)$

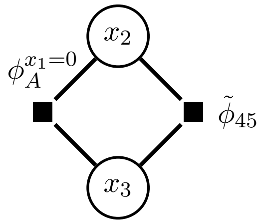
Solution. Starting with the factor graph for $p(x_2, x_3, x_4, x_5 \mid x_1 = 0, x_6 = 1)$



Marginalising x_4 combines the three factors ϕ_B, ϕ_C and $\phi_D^{x_6=1}$



Marginalising x_5 modifies the factor $\tilde{\phi}_4$



Marginalising x_3 combines the factors $\phi_A^{x_1=0}$ and $\tilde{\phi}_{45}$

We now compute the tables for the new factors $\tilde{\phi}_4$, $\tilde{\phi}_{45}$, $\tilde{\phi}_{453}$.
First find $\tilde{\phi}_4(x_2, x_3, x_5)$

x_2	x_3	x_4	ϕ_B
0	0	0	2
1	0	0	2
0	1	0	4
1	1	0	2
0	0	1	6
1	0	1	8
0	1	1	4
1	1	1	2

x_4	x_5	ϕ_C
0	0	8
1	0	2
0	1	2
1	1	6

x_4	$\phi_D^{x_6=1}$
0	6
1	3

so that $\phi_*(x_2, x_3, x_4, x_5) = \phi_B(x_2, x_3, x_4)\phi_C(x_4, x_5)\phi_D^{x_6=1}(x_4)$ equals

x_2	x_3	x_4	x_5	$\phi_*(x_2, x_3, x_4, x_5)$
0	0	0	0	2 * 8 * 6
1	0	0	0	2 * 8 * 6
0	1	0	0	4 * 8 * 6
1	1	0	0	2 * 8 * 6
0	0	1	0	6 * 2 * 3
1	0	1	0	8 * 2 * 3
0	1	1	0	4 * 2 * 3
1	1	1	0	2 * 2 * 3
0	0	0	1	2 * 2 * 6
1	0	0	1	2 * 2 * 6
0	1	0	1	4 * 2 * 6
1	1	0	1	2 * 2 * 6
0	0	1	1	6 * 6 * 3
1	0	1	1	8 * 6 * 3
0	1	1	1	4 * 6 * 3
1	1	1	1	2 * 6 * 3

and

x_2	x_3	x_5	$\sum_{x_4} \phi_B(x_2, x_3, x_4)\phi_C(x_4, x_5)\phi_D^{x_6=1}(x_4)$		$\tilde{\phi}_4$
0	0	0	(2 * 8 * 6) + (6 * 2 * 3)	=	132
1	0	0	(2 * 8 * 6) + (8 * 2 * 3)	=	144
0	1	0	(4 * 8 * 6) + (4 * 2 * 3)	=	216
1	1	0	(2 * 8 * 6) + (2 * 2 * 3)	=	108
0	0	1	(2 * 2 * 6) + (6 * 6 * 3)	=	132
1	0	1	(2 * 2 * 6) + (8 * 6 * 3)	=	168
0	1	1	(4 * 2 * 6) + (4 * 6 * 3)	=	120
1	1	1	(2 * 2 * 6) + (2 * 6 * 3)	=	60

Next find $\tilde{\phi}_{45}(x_2, x_3)$

x_2	x_3	x_5	$\tilde{\phi}_4$
0	0	0	132
1	0	0	144
0	1	0	216
1	1	0	108
0	0	1	132
1	0	1	168
0	1	1	120
1	1	1	60

so that

x_2	x_3	$\sum_{x_5} \tilde{\phi}_4(x_2, x_3, x_5)$		$\tilde{\phi}_{45}$
0	0	132 + 132	=	264
1	0	144 + 168	=	312
0	1	216 + 120	=	336
1	1	108 + 60	=	168

Finally find $\tilde{\phi}_{453}(x_2)$

x_2	x_3	$\phi_A^{x_1=0}$
0	0	4
1	0	2
0	1	2
1	1	6

x_2	x_3	$\tilde{\phi}_{45}$
0	0	264
1	0	312
0	1	336
1	1	168

so that

x_2	$\sum_{x_3} \tilde{\phi}_{45}(x_2, x_3)\phi_A^{x_1=0}(x_2, x_3)$		$\tilde{\phi}_{453}$
0	(4 * 264) + (2 * 336)	=	1728
1	(2 * 312) + (6 * 168)	=	1632

The normalising constant is $Z = 1728 + 1632$. Our conditional marginal is thus:

$$p(x_2 \mid x_1 = 0, x_6 = 1) = \begin{pmatrix} 1728/Z \\ 1632/Z \end{pmatrix} = \begin{pmatrix} 0.514 \\ 0.486 \end{pmatrix} \quad \text{(S.96)}$$

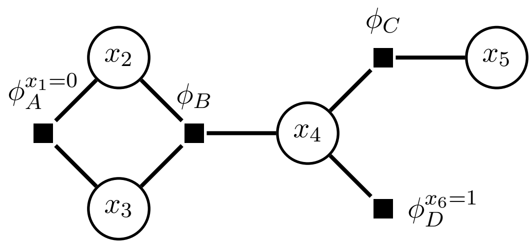
(c) *Now determine $p(x_2 \mid x_1 = 0, x_6 = 1)$ with the elimination ordering (x_5, x_4, x_3):*

(i) *Draw the graph for $p(x_2, x_3, x_4, \mid x_1 = 0, x_6 = 1)$ by marginalising x_5*
Compute the table for the new factor $\tilde{\phi}_5(x_4)$

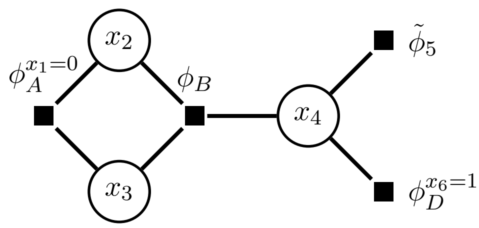
(ii) *Draw the graph for $p(x_2, x_3 \mid x_1 = 0, x_6 = 1)$ by marginalising x_4*
Compute the table for the new factor $\tilde{\phi}_{54}(x_2, x_3)$

(iii) *Draw the graph for $p(x_2 \mid x_1 = 0, x_6 = 1)$ by marginalising x_3*
Compute the table for the new factor $\tilde{\phi}_{543}(x_2)$

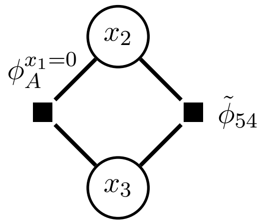
Solution. Starting with the factor graph for $p(x_2, x_3, x_4, x_5 \mid x_1 = 0, x_6 = 1)$



Marginalising x_5 modifies the factor ϕ_C



Marginalising x_4 combines the three factors ϕ_B, $\tilde{\phi}_5$ and $\phi_D^{x_6=1}$



Marginalising x_3 combines the factors $\phi_A^{x_1=0}$ and $\tilde{\phi}_{54}$

We now compute the tables for the new factors $\tilde{\phi}_5$, $\tilde{\phi}_{54}$, and $\tilde{\phi}_{543}$.
First find $\tilde{\phi}_5(x_4)$

x_4	x_5	ϕ_C
0	0	8
1	0	2
0	1	2
1	1	6

so that

x_4	$\sum_{x_5} \phi_C(x_4, x_5)$		$\tilde{\phi}_5$
0	$8 + 2$	=	10
1	$2 + 6$	=	8

Next find $\tilde{\phi}_{54}(x_2, x_3)$

x_2	x_3	x_4	ϕ_B
0	0	0	2
1	0	0	2
0	1	0	4
1	1	0	2
0	0	1	6
1	0	1	8
0	1	1	4
1	1	1	2

x_4	$\tilde{\phi}_5$
0	10
1	8

x_4	$\phi_D^{x_6=1}$
0	6
1	3

so that $\phi_*(x_2, x_3, x_4) = \phi_B(x_2, x_3, x_4)\tilde{\phi}_5(x_4)\phi_D^{x_6=1}(x_4)$ equals

x_2	x_3	x_4	$\phi_*(x_2, x_3, x_4)$
0	0	0	2 * 10 * 6
1	0	0	2 * 10 * 6
0	1	0	4 * 10 * 6
1	1	0	2 * 10 * 6
0	0	1	6 * 8 * 3
1	0	1	8 * 8 * 3
0	1	1	4 * 8 * 3
1	1	1	2 * 8 * 3

and

x_2	x_3	$\sum_{x_4} \phi_B(x_2, x_3, x_4)\tilde{\phi}_5(x_4)\phi_D^{x_6=1}(x_4)$		$\tilde{\phi}_{54}$
0	0	(2 * 10 * 6) + (6 * 8 * 3)	=	264
1	0	(2 * 10 * 6) + (8 * 8 * 3)	=	312
0	1	(4 * 10 * 6) + (4 * 8 * 3)	=	336
1	1	(2 * 10 * 6) + (2 * 8 * 3)	=	168

Finally find $\tilde{\phi}_{543}(x_2)$

x_2	x_3	$\phi_A^{x_1=0}$
0	0	4
1	0	2
0	1	2
1	1	6

x_2	x_3	$\tilde{\phi}_{54}$
0	0	264
1	0	312
0	1	336
1	1	168

so that

x_2	$\sum_{x_3} \tilde{\phi}_{54}(x_2, x_3)\phi_A^{x_1=0}(x_2, x_3)$		$\tilde{\phi}_{543}$
0	(4 * 264) + (2 * 336)	=	1728
1	(2 * 312) + (6 * 168)	=	1632

As with the ordering in the previous part, we should come to the same result for our conditional marginal distribution.The normalising constant is $Z = 1728 + 1632$, so that the conditional marginal is

$$p(x_2 \mid x_1 = 0, x_6 = 1) = \begin{pmatrix} 1728/Z \\ 1632/Z \end{pmatrix} = \begin{pmatrix} 0.514 \\ 0.486 \end{pmatrix} \tag{S.97}$$

(d) *Which variable ordering, (x_4, x_5, x_3) or (x_5, x_4, x_3) do you prefer?*

Solution. The ordering (x_5, x_4, x_3) is cheaper and should be preferred over the ordering (x_4, x_5, x_3) .

The reason for the difference in the cost is that x_4 has three neighbours in the factor graph for $p(x_2, x_3, x_4, x_5 \mid x_1 = 0, x_6 = 1)$. However, after elimination of x_5, which has only one neighbour, x_4 has only two neighbours left. Eliminating variables with more neighbours leads to larger (temporary) factors and hence a larger cost. We can see this from the tables that were generated during the computation (or numbers that we needed to add together): for the ordering (x_4, x_5, x_3), the largest table had 2^4 entries while for (x_5, x_4, x_3), it had 2^3 entries.

Choosing a reasonable variable ordering has a direct effect on the computational complexity of variable elimination. This effect becomes even more pronounced when the domain of our discrete variables has a size greater than 2 (binary variables), or if the variables are continuous.