Q: What is \( p(x_1) \)?

A: \[ \sum_{x_2} \sum_{x_3} p(x_1, x_2, x_3) \]

But:

\[
\sum_{x_2} \sum_{x_3} \left[ p(x_1, x_2, x_3) \right] = \frac{1}{Z} \sum_{x_2} \sum_{x_3} \left[ \phi_1(x_1) \cdot \phi_2(x_2) \cdot \phi_3(x_3) \cdot \psi(x_1, x_2) \cdot \psi(x_2, x_3) \right] = \frac{1}{Z} \phi_1(x_1) \sum_{x_2} \left[ \phi_2(x_2) \cdot \psi(x_1, x_2) \right] \left[ \sum_{x_3} \phi_3(x_3) \cdot \psi(x_2, x_3) \right]
\]

"message" from 3 \( \rightarrow \) 2

"message" from 2 \( \rightarrow \) 1

Clunky, because becomes exponential in \# of vars.
So:

\[ p(x_1) = \frac{1}{Z} \phi(x_1). M_{21}(x_1) \]

\[ M_{21}(x_1) = \sum_{x_2} \phi(x_2). \psi(x_1, x_2) M_{32}(x_2) \]

\[ M_{32}(x_2) = \sum_{x_3} \phi(x_3). \psi(x_2, x_3) \]

But this is a marginal, so \( Z = \sum_{x_1} \phi(x_1) M_{21}(x_1) \)

So \( Z = \sum_{x_1} \phi(x_1) M_{21}(x_1) \)

\[ p(x_1) = \frac{1}{Z} \phi(x_1). \sum_{x_2} \left[ \phi(x_2). \psi(x_1, x_2) \right] \left[ \sum_{x_3} \phi(x_3). \psi(x_2, x_3) \right] \times \left[ \sum_{x_4} \phi(x_4). \psi(x_2, x_4) \right] \]
\[ p(x_i) = \frac{1}{2} \phi(x_i) \cdot m_{21}(x_i) \]

\[ m_{21}(x_i) = \sum_{x_2} \phi(x_2) \cdot \Psi(x_i, x_2) \cdot \prod_{j \in \mathcal{N}(i) - 1} m_{j2}(x_j) \]

\[ m_{32}(x_2) = \sum_{x_3} \phi(x_3) \cdot \Psi(x_2, x_3) \]

etc.

This gives a general process for finding marginals on trees.

\[ p(x_i) \propto \phi_i(x_i) \cdot \prod_{j \in \mathcal{N}(i)} m_{ji}(x_i) \]

\[ m_{ji}(x_i) \leftarrow \sum_{x_j} \phi_j(x_j) \cdot \Psi(x_i, x_j) \prod_{k \in \mathcal{N}(j) - i} m_{kj}(x_j) \]
Now choose a root.

- orient all edges toward root.
- compute rootward messages, leaf $\rightarrow$ root.
- now leatward messages, root $\rightarrow$ leaf.
- done - you have all msgs, so can compute any marginal on one var that amuses.
(and it is easy to see that $p(y_i | x_i)$ is a)

\[
\prod_{i \in N(i) - j} \prod_{j \in N(i) \setminus \{i\}} p(y_i | x_i, x_j)
\]

Subject to

\[
\sum_{\omega_A \in S_A} \sum_{\omega_B \in S_B} p(\omega_A, \omega_B | \omega_i, x_i) = \sum_{\omega_A \in S_A} \sum_{\omega_B \in S_B} p(\omega_A, \omega_B) = 1
\]

Marginals

\[
p(x_i | \omega_i)
\]
It is straightforward to do MAP inference like this, by modifying messages. 

\[
\begin{align*}
\max_{x_1, x_2, x_3} &\quad p(x_1, x_2, x_3) \\
&= \max \frac{1}{Z} \phi_1(x_1) \psi(x_1, x_2) \phi(x_2) \psi(x_2, x_3) \phi(x_3) \\
&= \max_{x_1, x_2} \left[ \phi_1(x_1) \psi(x_1, x_2) \phi(x_2) \right] \left\{ \max_{x_3} \psi(x_2, x_3) \phi(x_3) \right\} \\
\end{align*}
\]

Not the same as previous message (max vs sum).
So we get:

\[
\text{MAP } x_i = \text{arg max } \left[ \phi_i(x_i), \prod_{u \in N(i)} M_{ui}(x_i) \right]
\]

\[
\text{MAP } M_{ji}(x_i) \leftarrow \text{max } x_j \left[ \phi_j(x_j), \Psi(x_i, x_j), \prod_{k \in N(j)-i} M_{kj}(x_j) \right]
\]

again, tree works great

- pass from leaves to root to
  compute \( M_{\text{MAP}} \)'s

- ... root to leaves to
  get \( x \)'s.
Idea:

- For non-trees, compute messages anyhow, iterate until it all settles down.
- Loopy belief propagation
- Often rather successful, but what does it mean?