

第二章 基础理论

本章内容

➤ 第二章 基础理论

➤ 2.1 信道编码定理

➤ 2.2 硬判决与软判决

➤ 2.3 基本信道模型及其信道容量

➤ 2.4 MAP与ML算法

➤ 2.5 因子图与和积算法

2.1 信道编码定理

➤ 信道编码定理：对于一个有噪信道，信道容量为C，只要数据传输速率 $R < C$ ，总会存在一种编码方法，使编码错误概率p随着码长n的增加，按指数下降到任意小的值（用最大似然译码）。即可以通过编码使通信过程实际上不发生错误，或使错误控制在允许的数值下。

➤ 根据信息论，信道容量是由输入和输出的最大互信息量决定的，即
$$C = \max_{p(x)} I(X, Y)$$

其中X和Y分别代表信道的输入和输出； $p(x)$ 是X的概率密度函数； $I(X, Y)$ 为变量X和Y的互信息，其定义将根据具体信道类型（BSC、AWGN等）的不同有所区别。

2.2 硬判决和软判决

- 假设 \mathbf{X} 和 \mathbf{Y} 分别为 q 元符号和 Q 元符号，对于离散信道，输入变量 \mathbf{X} 和输出变量 \mathbf{Y} 之间的互信息 $I(\mathbf{X}, \mathbf{Y})$ 定义为：

$$I(X, Y) = \sum_{j=0}^{q-1} \sum_{i=0}^{Q-1} p(x_j) p(y_i | x_j) \log_2 \frac{p(y_i | x_j)}{p(y_i)}$$

对上式取最大值，就得到硬判决的信道容量 C_{hard} 。当输入输出均为二元离散符号时，信道就可以用BSC模型来描述。

- 若系统采用软判决译码，信道译码器的输入为连续值，即 $y \in (-\infty, +\infty)$ ，则信道等效为离散输入连续输出信道， \mathbf{X} 和 \mathbf{Y} 之间的互信息 $I(\mathbf{X}, \mathbf{Y})$ 为：

$$I(X, Y) = \sum_{j=0}^{q-1} \int_{-\infty}^{+\infty} p(x_j) p(y | x_j) \log_2 \frac{p(y | x_j)}{p(y)} dy$$

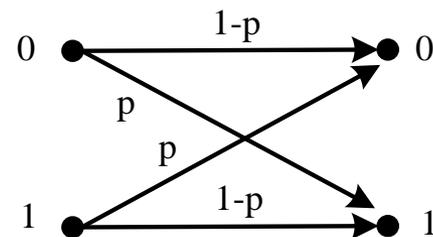
其中 $p(y|x_j)$ 表示发送 x_j 时解调器输出 y 的概率密度函数。对上式取最大值，就得到软判决时的信道容量 C_{soft} 。

2.3 基本信道模型及其信道容量

2.3.1 BSC信道

➤ 设传错概率为 p ，二进制对称信道(BSC)模型为：

其信道容量为：



$$C = \max_{p(x)} I(X, Y) = \sum_{j=0}^1 \sum_{i=0}^1 p(x_j) p(y_i | x_j) \log_2 \frac{p(y_i | x_j)}{p(y_i)} = 1 + p \log_2(p) + (1-p) \log_2(1-p)$$

我们知道，对于一个集合 X ，其熵为：
$$H(X) = \sum_{x \in X} p(x) \log \frac{1}{p(x)}$$

因此，BSC的信道容量为：
$$C = \max_{p(x)} I(X, Y) = 1 - H(p)$$

即每个信息比特携带 $1-H(p)$ 的信息。若采用码率为 R 的编码，在传输错误概率为 p 时，每个码字比特携带的信息量为 $R(1-H(p))$ ，它不能超过信道容量，因此有：

$$R(1-H(p)) \leq C \Rightarrow R \leq \frac{C}{1-H(p)}$$

- 对于AWGN信道，采用BPSK调制时BSC信道传输错误概率 p 为：

$$p = Q\left(\sqrt{\frac{2E_s}{N_0}}\right) = Q\left(\sqrt{\frac{2RE_b}{N_0}}\right)$$

- 若要 $p \rightarrow 0$ ，则实际信噪比只需大于 $R \rightarrow 0$ 时的信噪比（临界点）即可。

$$\lim_{p \rightarrow 0} (1 - H(p)) = \lim_{R \rightarrow 0} \frac{C}{R} = \lim_{R \rightarrow 0} \frac{1 + p \log_2(p) + (1-p) \log_2(1-p)}{R} \left(\begin{array}{l} \text{进行求导} \\ \ln(p) \text{的导数为 } p'/p \end{array} \right)$$

$$= \lim_{R \rightarrow 0} \left\{ p' \log_2(p) + p \frac{1}{(\ln 2)p} p' - p' \log_2(1-p) + (1-p) \frac{1}{(\ln 2)(1-p)} (-p') \right\}$$

- 由于 $\lim_{R \rightarrow 0} p = \lim_{R \rightarrow 0} Q\left(\sqrt{\frac{2RE_b}{N_0}}\right) = \frac{1}{2}$

- 错误概率 p 是一个Q函数， $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{-y^2/2} dy$ ，其导数为：

$$p' = -\frac{1}{2} \sqrt{\frac{E_b}{\pi RN_0}} e^{-\frac{RE_b}{N_0}}$$

➤ 这样 $\lim_{R \rightarrow 0} \frac{C}{R} = \frac{2}{\pi \ln(2)} \frac{E_b}{N_0}$

实现无差错传输时， $p=0$ ，从而 $H(b)=0$ ，因此

$$SNR = \frac{E_b}{N_0} = \frac{\pi \ln(2)}{2} = 0.37dB$$

即只有在信道SNR大于0.37dB时才能实现无错传输，该值称为BSC信道的Shannon限。

2.3.2 AWGN信道的容量

➤ 连续输入、连续输出AWGN信道容量

➤ 带限AWGN信道的信道容量公式为：

$$C = W \log_2 \left(1 + \frac{S}{N} \right) = W \log_2 \left(1 + \frac{S}{N_0 W} \right)$$

其中W为信道带宽，噪声的双边功率谱密度为 $N_0/2$ 。

当编码速率为R，信息比特能量为 E_b 时，有

$$\frac{C}{W} = \log_2 \left(1 + \frac{R E_b}{N_0 W} \right)$$

当 $W \rightarrow \infty$ 时，得到Shannon限的渐近值为：

$$SNR = \frac{E_b}{N_0} \geq \lim_{W \rightarrow \infty} \frac{2^{R/W} - 1}{C/W} = \ln 2 = -1.6dB$$

即只有当SNR不小于-1.6dB时才可能实现无错传输，这就是理想情况下AWGN信道的Shannon限。

- 对于带限信道上的二元符号而言，在满足Nyquist的理想情况下（ $W = 1/2T$ ）， $S = RE_b/T$ ， T 为符号周期，AWGN信道的信道容量（每符号所承载的信息量）为：

$$C' = C \cdot T = W \cdot T \log_2 \left(1 + \frac{S}{N} \right) = \frac{1}{2} \log_2 \left(1 + \frac{RE_b}{N_0 WT} \right) = \frac{1}{2} \log_2 \left(1 + \frac{2RE_b}{N_0} \right)$$

则由 $R \leq C' = \frac{1}{2} \log_2 \left(1 + \frac{2RE_b}{N_0} \right)$ ，可得

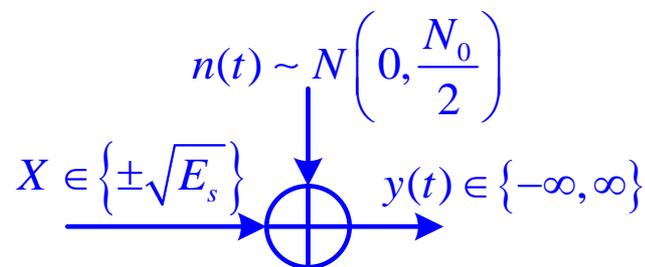
$$\frac{E_b}{N_0} \geq \frac{2^{2R} - 1}{2R}$$

当 $R = 1/2$ 时，Shannon容量限为0dB。

➤ 输入离散、输出连续AWGN信道容量

- 输入为二元符号，电平幅度为 $\pm\sqrt{E_s}$ ，与BSC类似，只有在输入符号等概时才能使互信息值达到最大，即信道容量。此时输入符号的概率密度函数为 $p(x) = \frac{1}{2} \cdot \left(\delta(x + \sqrt{E_s}) + \delta(x - \sqrt{E_s}) \right)$ 。
- 设 $y(t) = x(t) + n(t)$ ，其中 $n(t)$ 服从均值为0、方差为 $\sigma^2 = N_0/2$ 的高斯分布， $x(t) = \pm\sqrt{E_s}$ ，条件概率密度函数为：

$$p(y | x = \pm\sqrt{E_s}) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y \mp \sqrt{E_s})^2}{2\sigma^2}\right)$$



AWGN信道模型

- 根据信道容量定义，有：

$$C = \max_{p(x)} I(X, Y) = \max_{p(x)} (H(X) - H(X | Y))$$

$$H(X) = -\sum_x p(x) \log_2 p(x) = 1$$

$$H(X|Y) = - \sum_{x=\pm\sqrt{E_s}} \int_{-\infty}^{+\infty} p(x, y) \log_2 p(x|y) dy$$

$$= - \sum_{x=\pm\sqrt{E_s}} \int_{-\infty}^{+\infty} p(y|x) p(x) \log_2 \left(\frac{p(y|x)p(x)}{p(y)} \right) dy$$

$$C = 1 - \max_{p(x)} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{(y - \sqrt{RE_b})^2}{N_0}\right) \log_2 \left(1 + \exp\left(-\frac{4y\sqrt{RE_b}}{N_0}\right) \right) dy$$

➤ 由 $R \leq C$ 即可求得 E_b/N_0 的 Shannon 限。

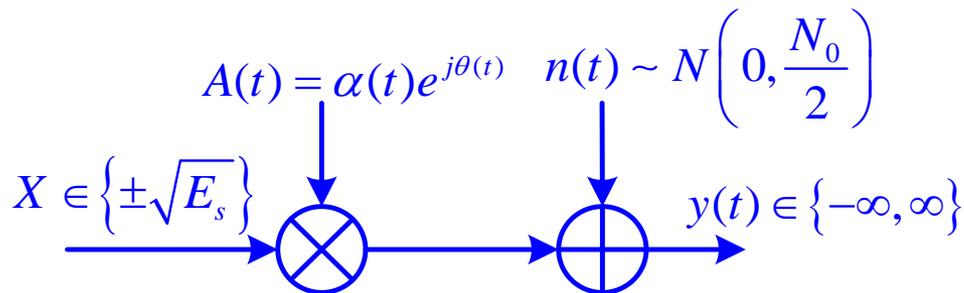
下表表示在不同编码速率、不同 BER 要求下，二元输入、连续输出 AWGN 信道下，系统所需的最小 SNR。

编码速率	$P_b(e)=10^{-2}$	$P_b(e)=10^{-3}$	$P_b(e)=10^{-4}$	$P_b(e)=10^{-5}$	$P_b(e)=0$
1/2	-0.357	0.112	0.117	0.185	0.187
1/3	-0.961	-0.559	-0.502	-0.496	-0.495

2.3.3 Rayleigh衰落信道的容量

- 在无线环境中，接收信号中包含很多反射波，根据中心极限定理，信道冲激响应可看作是两个正交的0均值、 σ_s 方差的高斯分量的组合，因此，信号的包络服从Rayleigh分布，相位服从 $\{-\pi, +\pi\}$ 的均匀分布。信号在Rayleigh衰落信道中传输会引入乘性干扰和加性噪声，接收信号为：

$$y(t) = A(t)x(t) + n(t)$$



其中 $A(t) = \alpha(t)e^{j\theta(t)}$ 为乘性干扰， $\alpha(t)$ 称为信道增益，若仅仅考虑信号的包络，则接收信号可表示为 $y = \alpha x + n$ ，其概率密度函数为Rayleigh分布，即

$$p(\alpha) = \frac{\alpha}{\sigma_s^2} e^{-\alpha^2/\sigma_s^2}, \quad \alpha \geq 0$$

式中 σ_s 为高斯分布的方差，由上式可得Rayleigh分布的均值 m_α 和方差 σ_α^2 分别为

$$m_\alpha = \sqrt{\pi/2} \sigma_s \quad \sigma_\alpha^2 = \left(2 - \frac{\pi}{2}\right) \sigma_s^2$$

- 将乘性因子归一化，即 $E\{\alpha^2\}=1$ ，则其分布可表示为：

$$p(\alpha) = 2\alpha e^{-\alpha^2}, \quad \alpha \geq 0$$

- 若信道增益已知，则称有信道边信息(Side Information, SI)，否则，就称无信道边信息(NSI)。下面给出这两种情况下的信道容量

- 有信道边信息

$$C = -\int_{\alpha} \int_y p(\alpha) p(y | x = \sqrt{E_s}, \alpha) \log \left[\frac{1}{2} (1 + \psi(y, \alpha)) \right] dy d\alpha$$

其中 $p(\alpha)$ 归一化的Rayleigh分布的PDF， $p(y|x, \alpha)$ 是均值为 α ，方差为 σ^2 的高斯PDF， $\psi(y, \alpha) = \exp(-2\alpha y / \sigma^2)$ 。容量可通过数值积分得到。

- 无信道边信息

$$C = -\int_{\alpha} \int_y p(\alpha) p(y | x = \sqrt{E_s}, \alpha) \log \left[\frac{1}{2} (1 + \psi(y)) \right] dy d\alpha$$

$$\text{其中 } \psi(y) = \frac{\int_{\alpha} p(\alpha) p(y | x = -\sqrt{E_s}, \alpha) d\alpha}{\int_{\alpha} p(\alpha) p(y | x = +\sqrt{E_s}, \alpha) d\alpha} \quad \text{容量通过数值积分得到。}$$

E. K. Hall, S.G. Wilson. Design and analysis of Turbo codes on Rayleigh fading channels. IEEE JSAC, Vol.16, No.2, 1998. p160~174.

2.4 MAP与ML算法

➤ 设 s 表示发送端的编码序列，其中 $s \in S$ 的选择概率为 $p(s)$ ，接收机用接收到的序列 r 来判决发送信号 s 是什么。设估计的判决 $\hat{s} \in S$ 。

➤ 定理：最小化错误概率的判决准则是选择能使 $p(s|r)$ 最大的那个 s ，即

$$\hat{s} = \arg \max_{s \in S} p(s | r)$$

➤ 根据Bayes准则，上式可写成

$$\hat{s} = \arg \max_{s \in S} p(s | r) = \arg \max_{s \in S} \frac{p(r | s)p(s)}{p(r)}$$

➤ 由于分母不依赖于 s ，上式可进一步写成

$$\hat{s} = \arg \max_{s \in S} p(r | s)p(s)$$

最大后验概率准则

如果所有先验概率是相同的，该

准则可简化为：

$$\hat{s} = \arg \max_{s \in S} p(r | s)$$

最大似然判决准则

如果所有先验概率是相同的，两者是等价的！

思考一下：最大似然准则在什么情况下不是最优算法，为什么？

2.5 因子图与和积算法

- 在1981年Tanner的论文中，介绍了一种可以用来表示码字的图形，称为Tanner图。Tanner图包含两类节点：码元（变量）节点和校验节点，然后通过边连接这两种不同的节点，并且同种节点间不能有直接的边连接。如果给定一个码字的码元数和它的校验方程，则用Tanner图可以唯一地确定该码字。例如一个(7,3)线性分组码，其校验方程为：

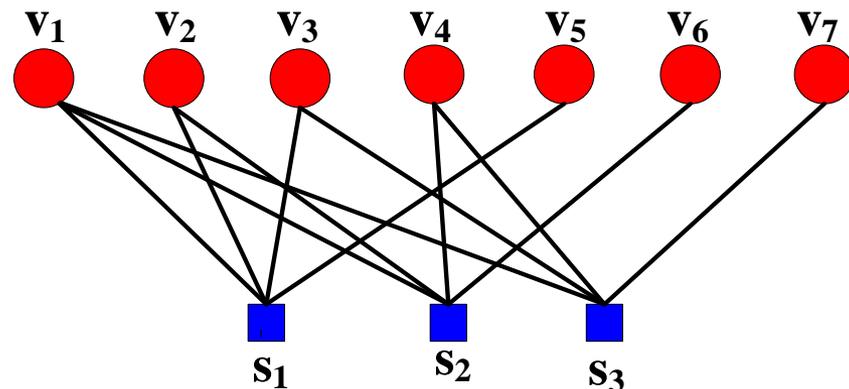
$$H = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 & \mathbf{v}_6 & \mathbf{v}_7 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \end{matrix}$$

$\mathbf{v}_1 \sim \mathbf{v}_7$ 是变量节点，

$\mathbf{s}_1 \sim \mathbf{s}_3$ 是校验节点。

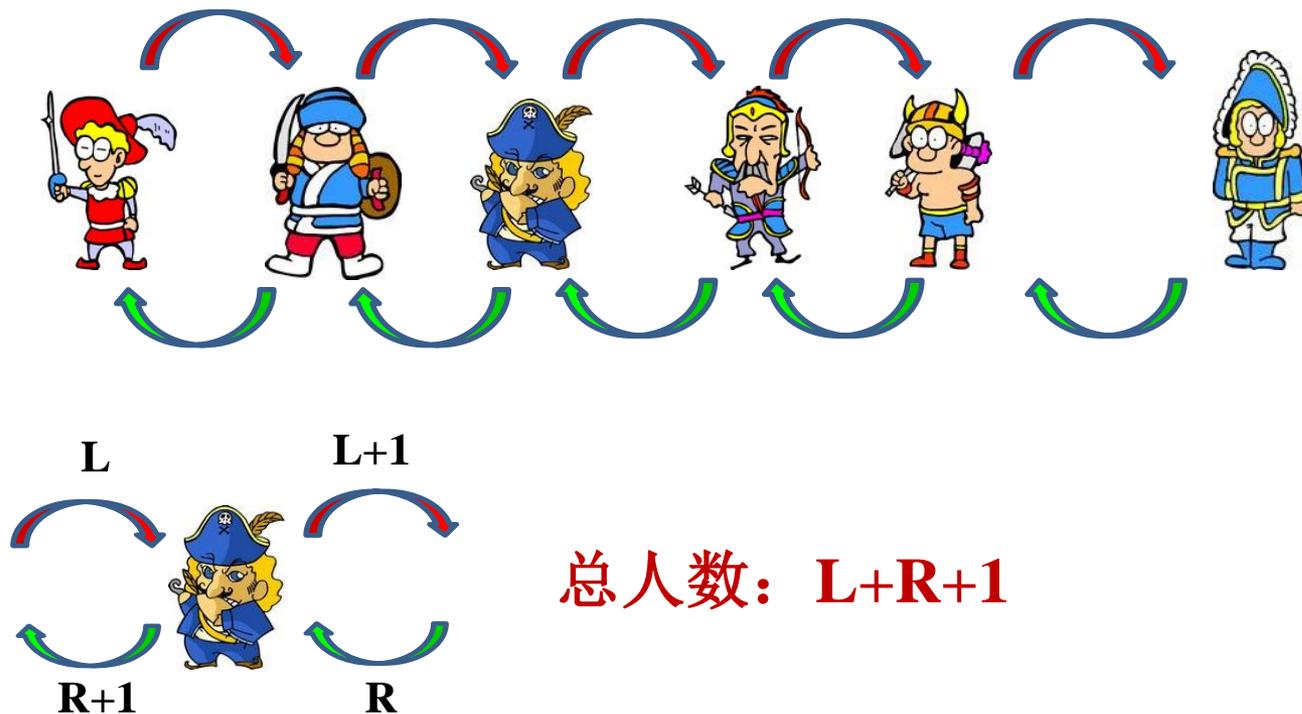
在用Tanner图表示的过程中，码的译码算法是可以明确的公式反映并可程式化进行，即有很强的可实现性。

Tanner图从本质上讲是用码字的校验方程来表述码字。



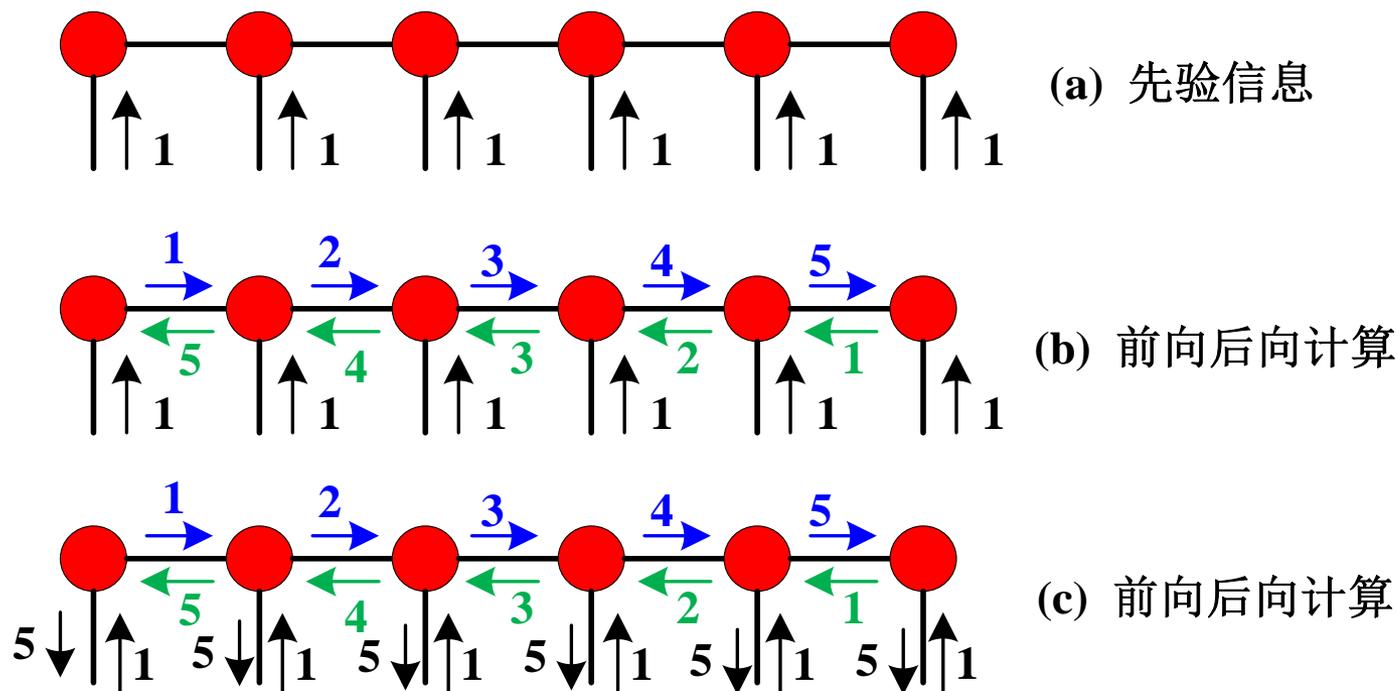
2.5.1 消息传递算法

- 和积算法 (Sum-Product Algorithm) 作为一种通用的消息传递算法 (Message Passing Algorithm), 描述了因子图中顶点 (变量节点和校验节点) 处的信息计算公式, 而在基于图的编译码系统中, 我们首先需要理解的是顶点之间是如何通过边来传递信息。
- 例如, 若干士兵排成一队列, 每个士兵只能与他相邻的士兵交流, 问如何才能使每个士兵都知道总的士兵数?



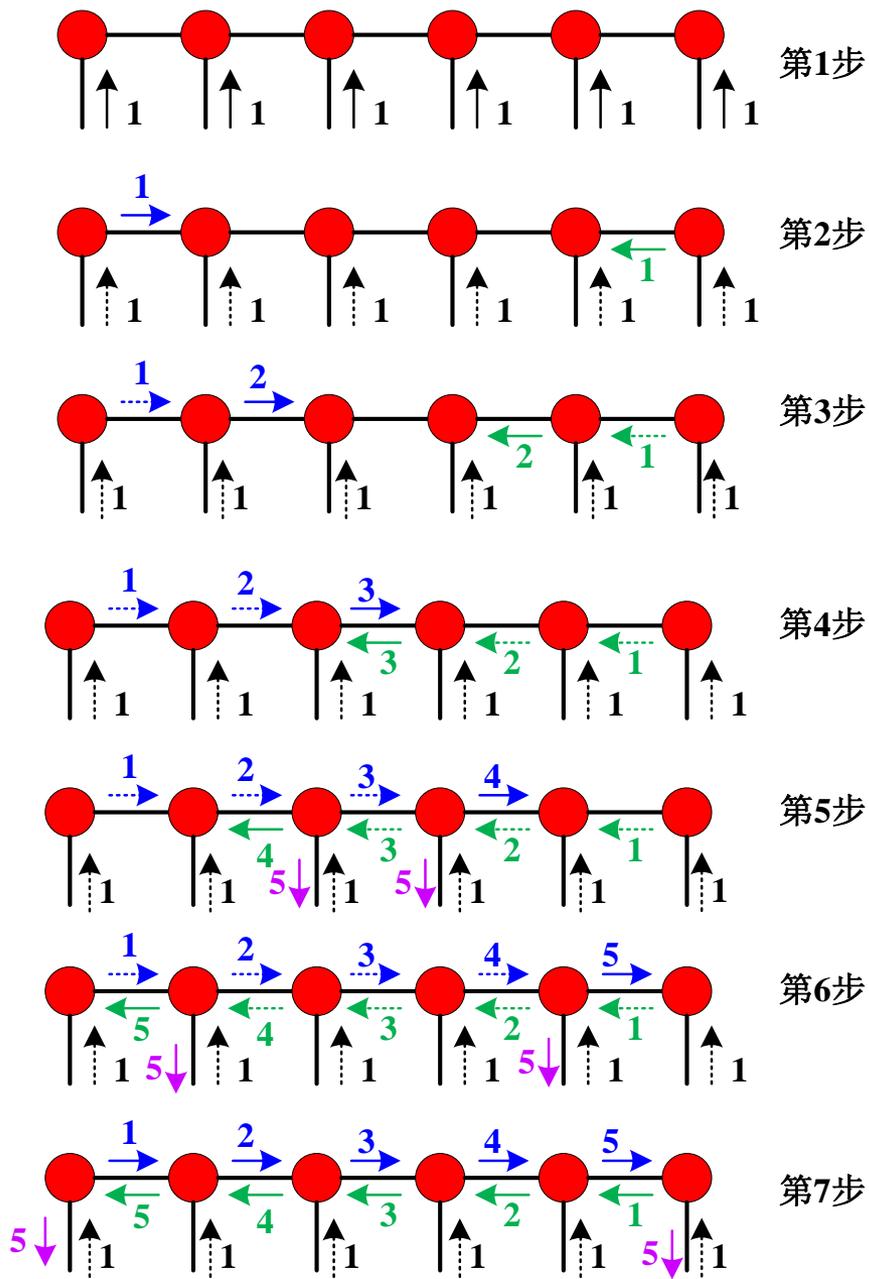
总人数: $L+R+1$

- 将上面的情况抽象成一般模型，如下图所示，图中每个点可看作是一个士兵，对应到编码理论中则可看作因子图中对应的节点（如前述的变量节点和校验节点，或卷积码的BCJR算法中的状态点）。



对应上图，引入几个概念：**先验信息P**、**外信息E**以及**后验信息A**。在上例中，先验信息P表示每个士兵自身的数字1，如图（a）所示；外信息E表示从其他相邻的士兵获取的信息，如图（c）所示，即每个士兵的外信息均为5；后验信息 $A = P + E$ ，在这里表示队列的总人数，即为6。从图中可以看出，得到最后的结果是通过**前向计算**和**后向计算**得到的。

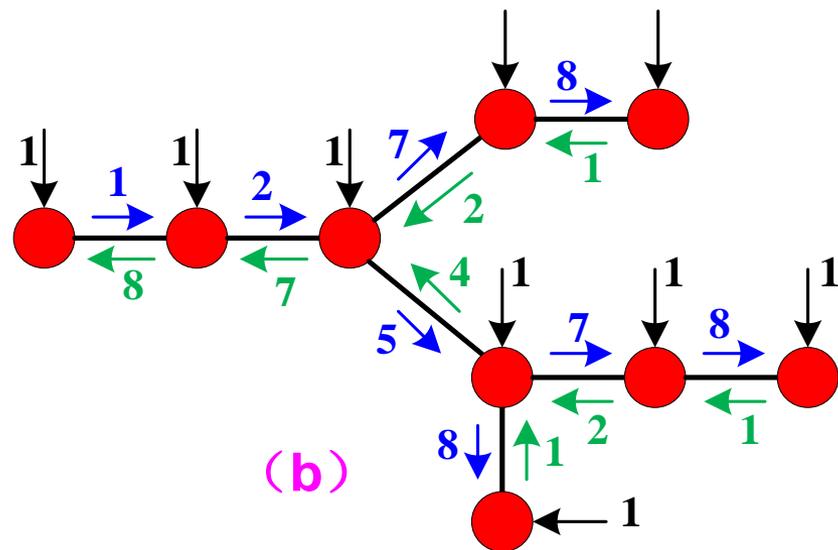
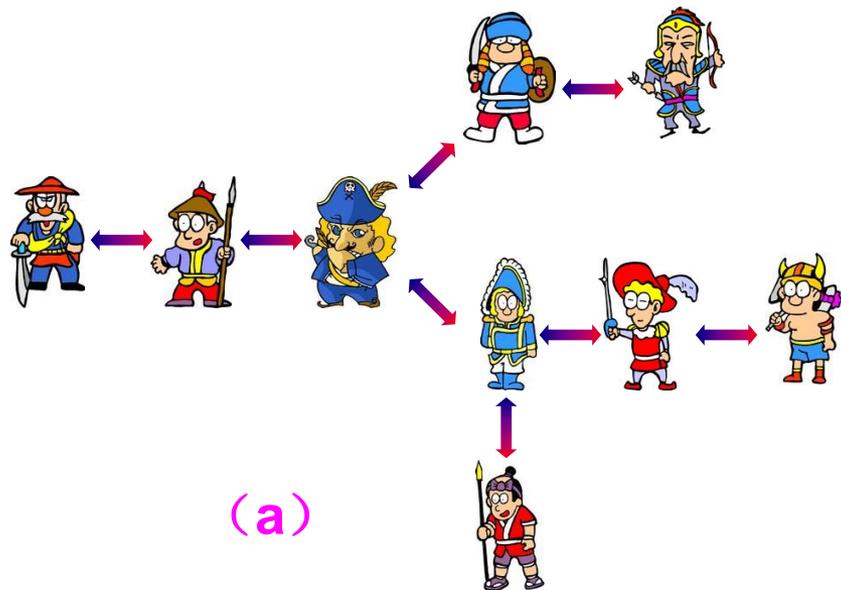
两个方向同时计算的前向 / 后向消息传递过程



➤ 从图中可以看出，经过5步计算第3、第4个士兵可以得到总人数 $6(5+1)$ ，经过6步计算第2、第5个士兵可以得到总人数，经过7步计算所有士兵均知道了总人数。



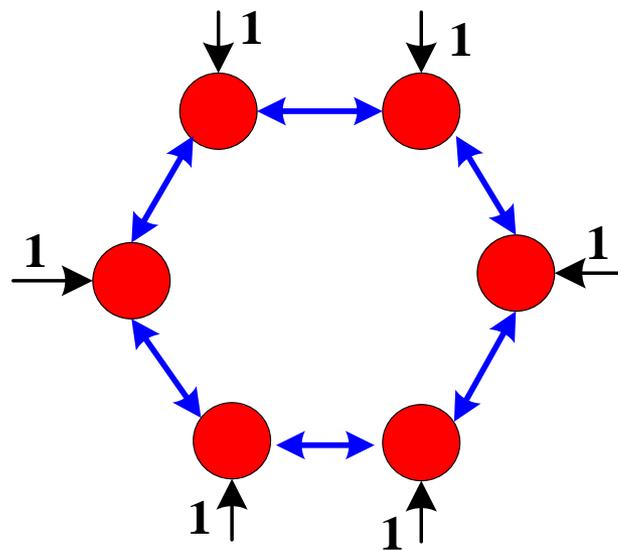
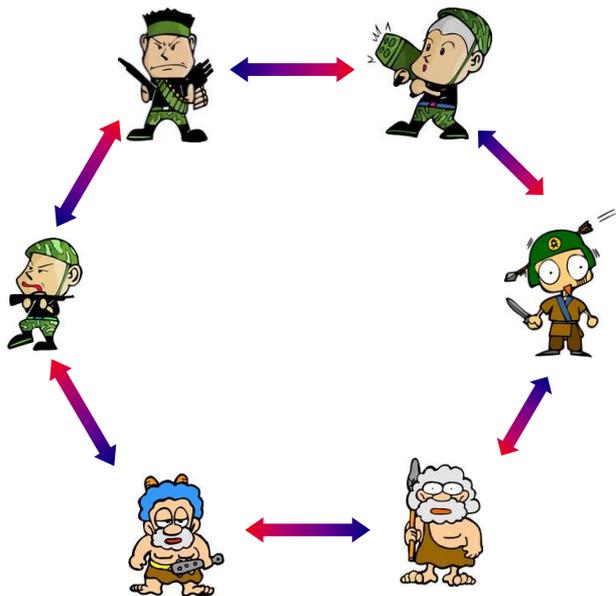
问题：如果士兵不是按照队列排列，怎么办呢？



- 该图(a)与前图不同之处在于，有些士兵不止有2个相邻的士兵，可能有3个或更多。具体信息传递流程如图(b)所示。每个士兵同样可以获得其相邻士兵给他的外信息，同时加上自身的信息然后传递给相邻的士兵。
- 在上面的两个例子中，每个士兵节点的信息只需在所有其相邻节点上进行一次前向和后向的计算，则每个士兵就可知道总人数。这样的图有一个共同特点：所有节点构成一棵树，而树结构中是没有环路的。



如果有环路，会导致什么结果？



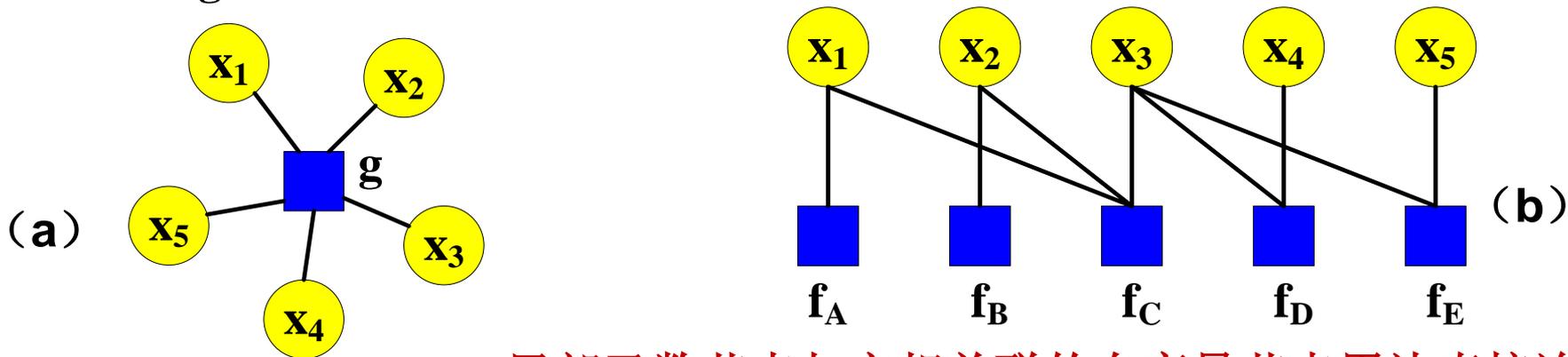
- 由于有环路的存在，如果用上述信息更新方法来确定总人数，将会导致无法确定何时中止信息的传递，因此也就无法确定士兵人数。对应到编码理论，则在设计LDPC码的校验矩阵时，应尽量避免校验矩阵对应的因子图中出现短环（如4循环、6循环、8循环等）。

2.5.2 因子图

- 将前述消息传递算法中的节点构成的图(Tanner图)更一般化就得到了因子图。因子图是一种用于描述多变量函数的“二部图”(Bipartite Graph)。一般来说，在因子图中存在两类节点：**变量节点**和对应的**函数节点**，变量节点所代表的变量是函数节点的自变量。同类节点之间没有**边**直接相连。
- 例如：图(a)表示有5个自变量的函数g的因子图。假定函数g可以分解成几个“局部函数”之积的形式，即：

$$g(x_1, x_2, x_3, x_4, x_5) = f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_3, x_4) f_E(x_3, x_5)$$

则函数g的因子图表示如图(b)所示



局部函数节点与它相关联的自变量节点用边直接连接，它反映了全局函数g和局部函数之间的关系。

2.5.3 边缘函数

- 在上一页的图 (a) 中, 反映的是全局函数 $g(\dots)$, 但我们希望得到的往往是通过这个全局函数计算出的各个边缘函数, 如求上例中的 $\tilde{g}(x_1)$ 。

设:
$$\tilde{g}(x_1) = \sum_{x_2, x_3, x_4, x_5} g(x_1, x_2, x_3, x_4, x_5)$$

$\tilde{g}(x_1)$ 是函数 $g(x_1, x_2, x_3, x_4, x_5)$ 当变量为 $x_2 \sim x_5$ 时的和函数。将p21页的公式带入上式, 可写为:

$$\begin{aligned}\tilde{g}(x_1) &= \sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_3, x_4) f_E(x_3, x_5) \\ &= f_A(x_1) \sum_{x_2} f_B(x_2) \sum_{x_3} f_C(x_1, x_2, x_3) \sum_{x_4} f_D(x_3, x_4) \sum_{x_5} f_E(x_3, x_5)\end{aligned}$$

也可写为

$$= f_A(x_1) \sum_{\sim\{x_1\}} \left\{ f_B(x_2) f_C(x_1, x_2, x_3) \cdot \left(\sum_{\sim\{x_3\}} f_D(x_3, x_4) \right) \cdot \left(\sum_{\sim\{x_3\}} f_E(x_3, x_5) \right) \right\}$$

$$\sum_{\sim\{x_2\}} h(x_1, x_2, x_3) \triangleq \sum_{x_1 \in A_1} \sum_{x_3 \in A_3} h(x_1, x_2, x_3)$$

$$g(x_1, x_2, x_3, x_4, x_5) = f_A(x_1)f_B(x_2)f_C(x_1, x_2, x_3)f_D(x_3, x_4)f_E(x_3, x_5)$$

➤ 如果求 $\tilde{g}(x_3)$ ，即将 x_1, x_2, x_4, x_5 作为变量时， $g(x_1, x_2, x_3, x_4, x_5)$ 的和函数。

$$\begin{aligned}\tilde{g}(x_3) &= \sum_{x_1, x_2, x_4, x_5} g(x_1, x_2, x_3, x_4, x_5) \\ &= \left(\sum_{\sim\{x_3\}} f_A(x_1)f_B(x_2)f_C(x_1, x_2, x_3) \right) \cdot \left(\sum_{\sim\{x_3\}} f_D(x_3, x_4) \right) \cdot \left(\sum_{\sim\{x_3\}} f_E(x_3, x_5) \right)\end{aligned}$$

➤ 推广到一般情况， $\tilde{g}(x_i)$ 为：

$$\tilde{g}(x_i) = \sum_{x_j} g(x_1, x_2, \dots, x_i, \dots, x_N)$$

其中 $i \in \{1, 2, \dots, N\}$, $\forall j \in \{1, 2, \dots, N\} \setminus \{i\}$ 表示除去 i 的部分



计算复杂，如何简化呢？

首先进行一些数学定义：
$$f_I(x_3) = \sum_{x_5} f_E(x_3, x_5)$$

$$f_{II}(x_3) = \sum_{x_4} f_D(x_3, x_4)$$

$$f_{III}(x_3) = f_I(x_3) \cdot f_{II}(x_3)$$

可得到：
$$f_{IV}(x_1, x_2) = \sum_{x_3} f_C(x_1, x_2, x_3) f_{III}(x_3)$$

$$f_V(x_1) = \sum_{x_2} f_B(x_2) f_{IV}(x_1, x_2)$$

最终：
$$\tilde{g}(x_1) = f_A(x_1) \cdot f_V(x_1)$$

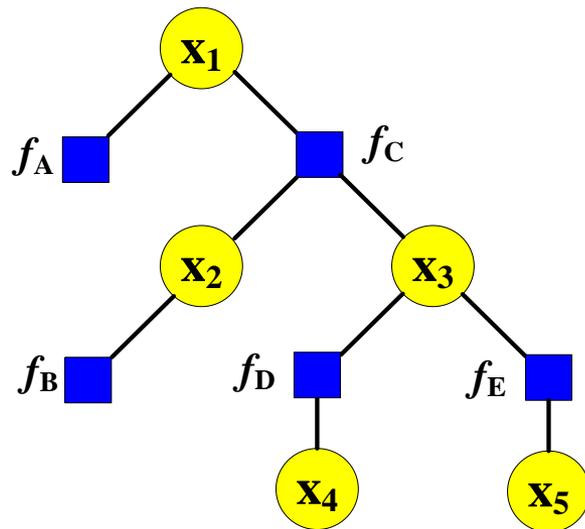
简化的思想是基于将一个复杂任务分解成多个小任务，每个小任务对应到因子图上就是一个函数节点。这使得其计算时不需要来自因子图其他部分的信息，且传送其计算结果仅由这些局部函数的自变量来承担，从而简化了计算。

2.5.4 和积算法的表示及工作原理

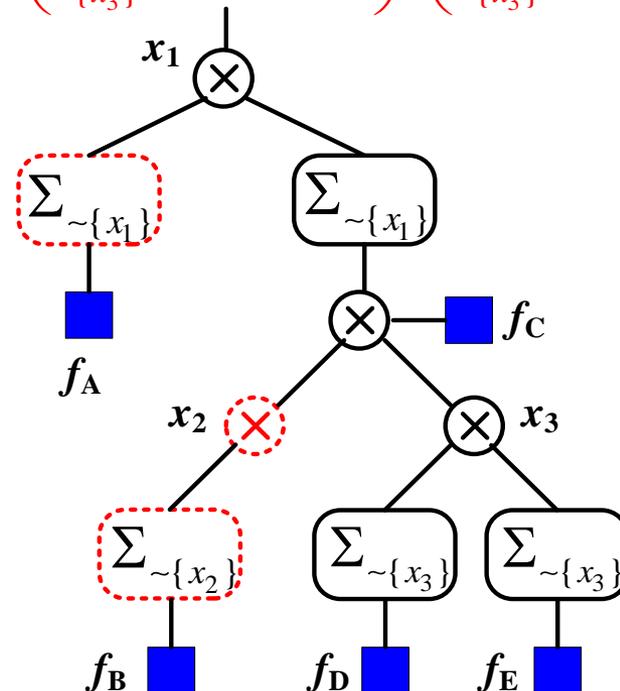
- 和积算法从本质上就是一种消息传递算法，它可从全局函数计算出各个不同的边缘函数。我们在前面求出了边缘函数 $\tilde{g}(x_1)$ ，其因子图表示如 (a) 所示。因子图只包含了变量与函数的对应关系，而运算关系并没有得到明确、具体的可视化，因此人们又提出了表达式树图 (expression trees) 的概念， $\tilde{g}(x_1)$ 的表达式树图如 (b) 所示。

$$g(x_1, x_2, x_3, x_4, x_5) = f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_3, x_4) f_E(x_3, x_5)$$

$$\tilde{g}(x_1) = f_A(x_1) \sum_{\sim\{x_1\}} \left\{ f_B(x_2) f_C(x_1, x_2, x_3) \cdot \left(\sum_{\sim\{x_3\}} f_D(x_3, x_4) \right) \cdot \left(\sum_{\sim\{x_3\}} f_E(x_3, x_5) \right) \right\}$$



(a)

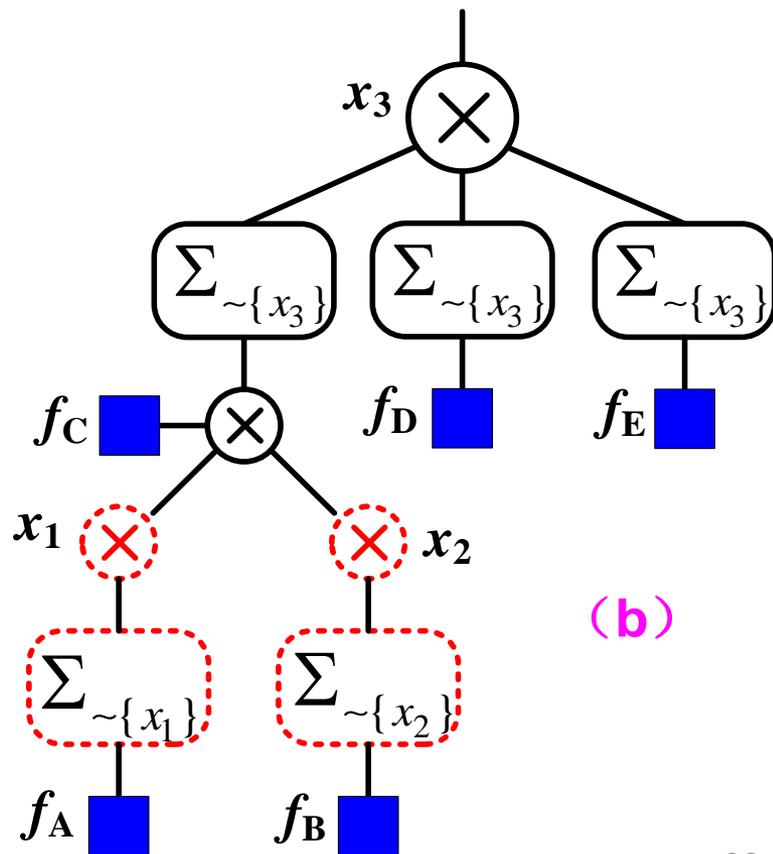
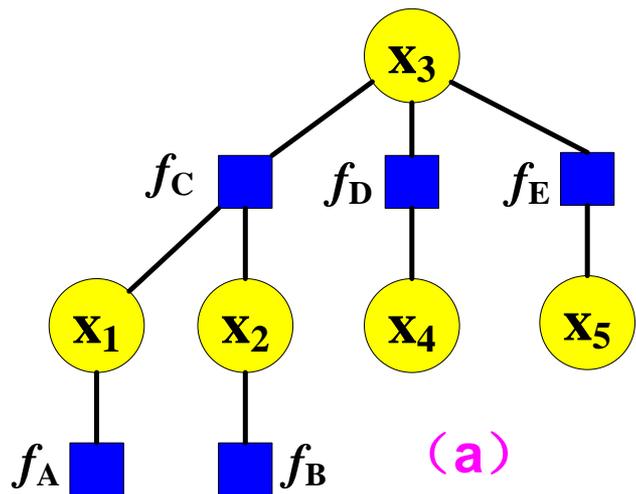


(b)

- 再例如，我们希望得到边缘函数 $\tilde{g}(x_3)$ 。根据下式，我们可以画出其因子图，如图 (a) 所示，其表达式树图如 (b) 所示。

$$g(x_1, x_2, x_3, x_4, x_5) = f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_3, x_4) f_E(x_3, x_5)$$

$$\tilde{g}(x_3) = \left(\sum_{\sim\{x_3\}} f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) \right) \cdot \left(\sum_{\sim\{x_3\}} f_D(x_3, x_4) \right) \cdot \left(\sum_{\sim\{x_3\}} f_E(x_3, x_5) \right)$$



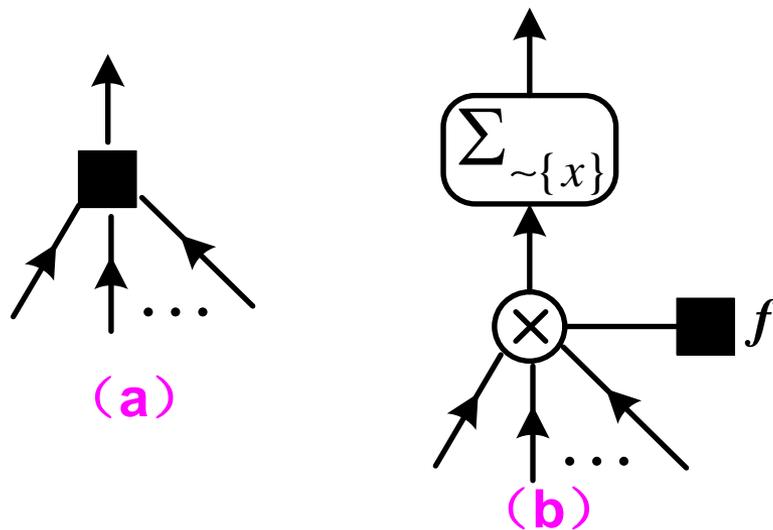
因子图和表达式树图的对应关系？

➤ 从表示全局函数的因子图到表示各边缘函数 $\tilde{g}(x_i)$ 的表达式树图的转换，应注意以下几点：

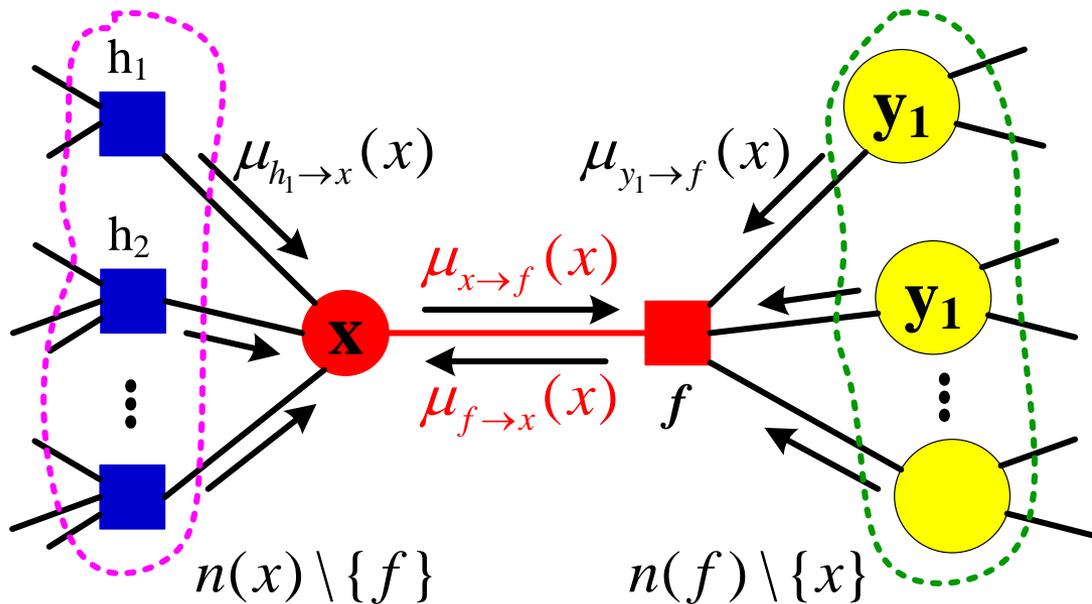
- ① 因子图中的每个变量用乘积运算符 “ \otimes ” 代替；
- ② 因子图中的函数节点用 “ \otimes —■ f ” 运算符代替（“form product and multiply f ”）；
- ③ 在函数节点 f 和其parent x 之间插入 “ $\sum_{\sim\{x\}}$ ” 和式运算符；

➤ 这些局部转换如下图所示。图（a）对应于变量节点，图（b）对应于有parent x 的函数节点。

没有操作对象的乘积运算符 “ \otimes ” 可当作乘以1对待，或当它处在表达式树图的末端时，可以忽略； 如果和式运算符 “ $\sum_{\sim\{x\}}$ ” 运用到只有一个自变量的函数中时，可以省略；



- 在消息传递的过程中，需要计算不同的和与积，因此又称为和积算法（Sum-Product Algorithm）。和积算法的更新如下图所示



设 $u_{x \rightarrow f}(x)$ 表示消息是从节点 x 到节点 f ， $u_{f \rightarrow x}(x)$ 表示消息是从节点 f 到节点 x ， $n(v)$ 表示在因子图中给定节点 v 的邻居集合。

变量到局部函数：

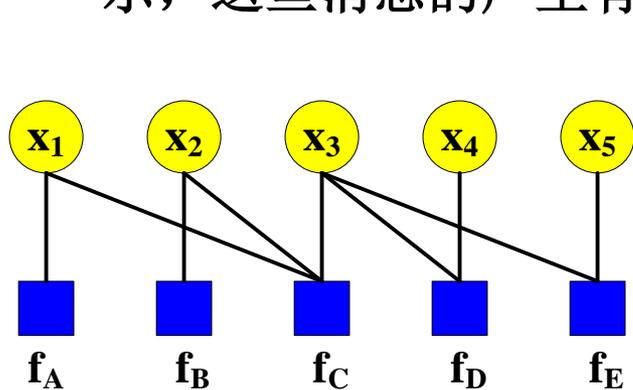
$$\mu_{x \rightarrow f}(x) = \prod_{h \in n(x) \setminus \{f\}} \mu_{h \rightarrow x}(x)$$

局部函数到变量：

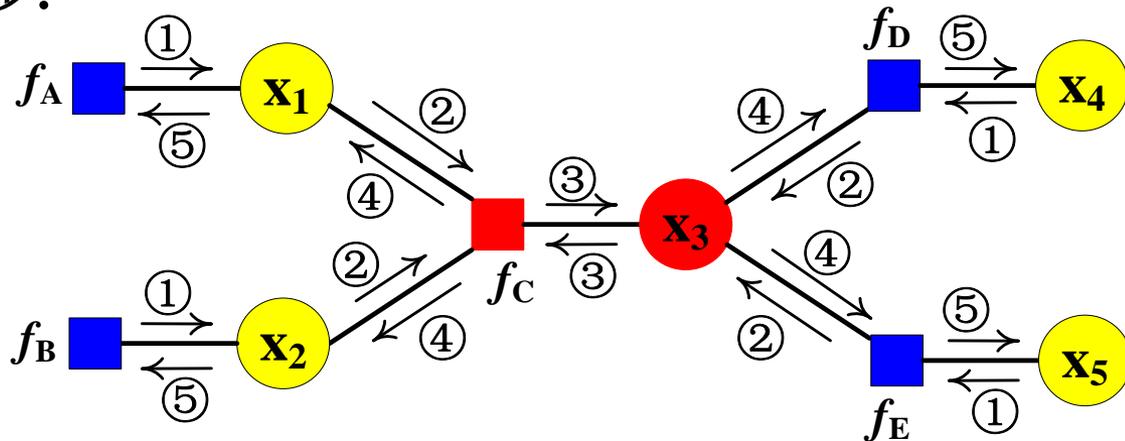
$$\mu_{f \rightarrow x}(x) = \sum_{\sim \{x\}} \left(f(X) \cdot \prod_{y \in n(f) \setminus \{x\}} \mu_{y \rightarrow f}(y) \right)$$

其中 $X = n(f)$ 是函数 f 的自变量集合

➤ 例：前面的因子图如图 (a) 所示，和积算法产生的消息流如图 (b) 所示，这些消息的产生有五步：



(a)



(b)

步骤1: $\mu_{f_A \rightarrow x_1}(x_1) = \sum_{\sim\{x_1\}} f_A(x_1) = f_A(x_1)$

$$\mu_{x_4 \rightarrow f_D}(x_4) = 1$$

$$\mu_{f_B \rightarrow x_2}(x_2) = \sum_{\sim\{x_2\}} f_B(x_2) = f_B(x_2)$$

$$\mu_{x_5 \rightarrow f_E}(x_5) = 1$$

步骤2:

$$\mu_{x_1 \rightarrow f_C}(x_1) = \mu_{f_A \rightarrow x_1}(x_1) \quad \mu_{f_D \rightarrow x_3}(x_3) = \sum_{\sim\{x_3\}} \left(f_D(x_3, x_4) \cdot \mu_{x_4 \rightarrow f_D}(x_4) \right)$$

$$\mu_{x_2 \rightarrow f_C}(x_2) = \mu_{f_B \rightarrow x_2}(x_2) \quad \mu_{f_E \rightarrow x_3}(x_3) = \sum_{\sim\{x_3\}} \left(f_E(x_3, x_5) \cdot \mu_{x_5 \rightarrow f_E}(x_5) \right)$$

步骤3:
$$\mu_{f_C \rightarrow x_3}(x_3) = \sum_{\sim\{x_3\}} \left(f_C(x_1, x_2, x_3) \cdot \mu_{x_1 \rightarrow f_C}(x_1) \cdot \mu_{x_2 \rightarrow f_C}(x_2) \right)$$

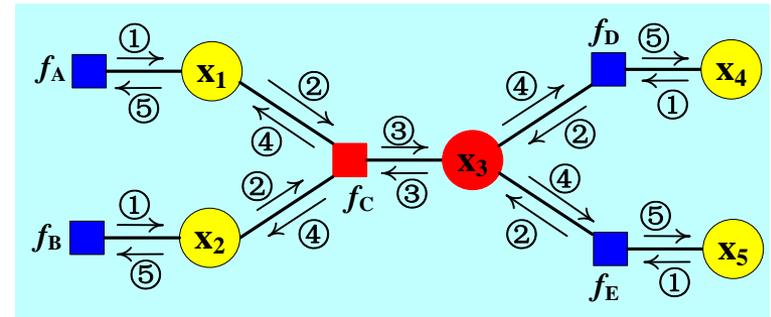
$$\mu_{x_3 \rightarrow f_C}(x_3) = \mu_{f_D \rightarrow x_3}(x_3) \cdot \mu_{f_E \rightarrow x_3}(x_3)$$

步骤4:
$$\mu_{f_C \rightarrow x_1}(x_1) = \sum_{\sim\{x_1\}} \left(f_C(x_1, x_2, x_3) \cdot \mu_{x_3 \rightarrow f_C}(x_3) \cdot \mu_{x_2 \rightarrow f_C}(x_2) \right)$$

$$\mu_{f_C \rightarrow x_2}(x_2) = \sum_{\sim\{x_2\}} \left(f_C(x_1, x_2, x_3) \cdot \mu_{x_3 \rightarrow f_C}(x_3) \cdot \mu_{x_1 \rightarrow f_C}(x_1) \right)$$

$$\mu_{x_3 \rightarrow f_D}(x_3) = \mu_{f_C \rightarrow x_3}(x_3) \cdot \mu_{f_E \rightarrow x_3}(x_3)$$

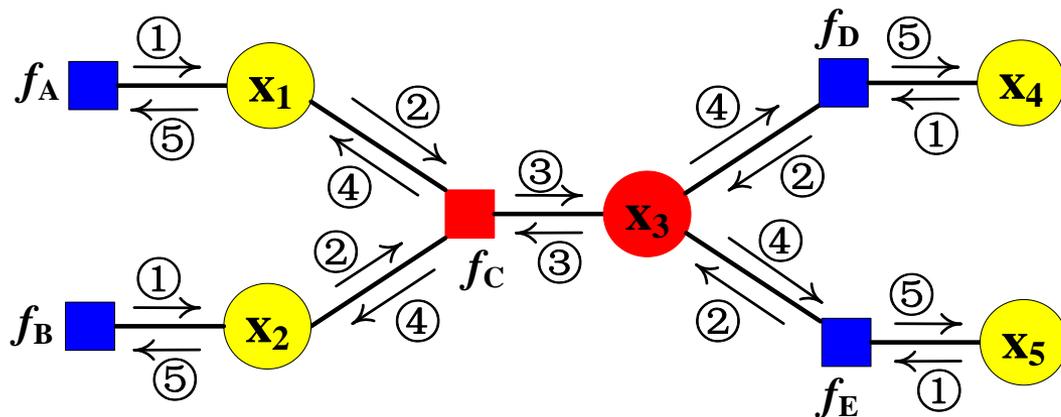
$$\mu_{x_3 \rightarrow f_E}(x_3) = \mu_{f_C \rightarrow x_3}(x_3) \cdot \mu_{f_D \rightarrow x_3}(x_3)$$



步骤5:

$$\mu_{x_1 \rightarrow f_A}(x_1) = \mu_{f_C \rightarrow x_1}(x_1) \quad \mu_{f_D \rightarrow x_4}(x_4) = \sum_{\sim\{x_4\}} \left(f_D(x_3, x_4) \cdot \mu_{x_3 \rightarrow f_D}(x_3) \right)$$

$$\mu_{x_2 \rightarrow f_B}(x_2) = \mu_{f_C \rightarrow x_2}(x_2) \quad \mu_{f_E \rightarrow x_5}(x_5) = \sum_{\sim\{x_5\}} \left(f_E(x_3, x_5) \cdot \mu_{x_3 \rightarrow f_E}(x_3) \right)$$



➤ 我们计算某个边缘函数 $\tilde{g}(x_i)$ ，就等效为送给 x_i 所有消息的乘积，因此就可得到每个边缘函数。

$$\tilde{g}(x_1) = \mu_{f_A \rightarrow x_1}(x_1) \cdot \mu_{f_C \rightarrow x_1}(x_1)$$

$$\tilde{g}(x_2) = \mu_{f_B \rightarrow x_2}(x_2) \cdot \mu_{f_C \rightarrow x_2}(x_2)$$

$$\tilde{g}(x_3) = \mu_{f_C \rightarrow x_3}(x_3) \cdot \mu_{f_D \rightarrow x_3}(x_3) \cdot \mu_{f_E \rightarrow x_3}(x_3)$$

$$\tilde{g}(x_4) = \mu_{f_D \rightarrow x_4}(x_4)$$

$$\tilde{g}(x_5) = \mu_{f_E \rightarrow x_5}(x_5)$$