TS345
-
Codage pour la 5G

Romain Tajan

13 octobre 2022

TS345 en bref...

Organisation du module

- 6 créneaux (1h20) de cours
- 3 créneaux de TP (2h40)

Découpage des cours

- 1 créneau de **rappels sur les codes correcteurs**
- 3 créneaux sur les **Codes LDPC**
- 2 créneaux sur les **Codes Polaires**

Plan

Plan

1. Introduction générale
   - ▷ Histoire de code correcteur
2. Rappels sur de codage / définitions
3. Codes Linéaires (binaires) en blocs
4. LDPC

Plan

1. Introduction générale
   - ▷ Histoire de code correcteur
2. Rappels sur de codage / définitions
3. Codes Linéaires (binaires) en blocs
4. LDPC

Un peu d'histoire...

1948 **Shannon** - capacité d'un canal (non constructive)

1955 **Elias** - Code convolutifs (GSM)

1960 **Reed et Solomon** - Codes RS (CD $\rightarrow$ BluRay, QR, DVB-S, RAID6)

Gallager - Codes LDPC

1966 **Forney** - Codes concatennés (Pioneer (1968-1972), Voyager (1977))

1967 **Viterbi** - Décodage optimal des codes convolutifs

1993 **Berrou, Glavieux et Thitimajshima** - Turbocodes (3G/4G, deep-space)

1996 **MacKay** - Ré-invente les LDPC (DVB-S2, WiFi, 5G)

2008 **Arikan** - Codes Polaires (5G)

Plan

Plan

Le canal...

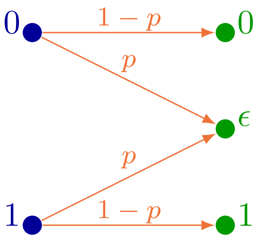
Un **canal** est défini par un triplet : $(\mathcal{X}, \mathcal{Y}, p(y|x))$ où

- $\mathcal{X}$ est l'**alphabet d'entrée**
- $\mathcal{Y}$ est l'**alphabet de sortie**
- $p(y|x)$ est la **probabilité de transition**

Soit $n \in \mathbb{N}$ et soit le canal $(\mathcal{X}^n, \mathcal{Y}^n, p(\mathbf{y}|\mathbf{x}))$, ce canal est dit "**sans mémoire**" si sa probabilité de transition vérifie

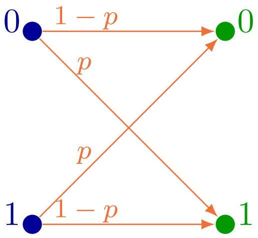
$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} p(y_i|x_i)$$

Le canal à effacement binaire



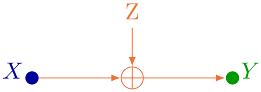
- $\mathcal{X} = \{0, 1\}$ (canal à entrées binaires)
- $\mathcal{Y} = \{0, \epsilon, 1\}$
- $p(\epsilon|0) = p(\epsilon|1) = p$ et $p(0|0) = p(1|1) = 1 - p$
- Canal utile pour les couches hautes, pour le stockage

Le canal binaire symétrique



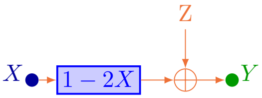
- $\mathcal{X} = \{0, 1\}$ (canal à entrées binaires)
- $\mathcal{Y} = \{0, 1\}$
- $p(1|0) = p(0|1) = p$ et $p(0|0) = p(1|1) = 1 - p$
- Canal utile après décision

Le canal additif gaussien



- $\mathcal{X} = \mathbb{R}$
- $\mathcal{Y} = \mathbb{R}$
- $p(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y-x)^2}$

Le canal additif gaussien à entrées binaires



- $\mathcal{X} = \{0, 1\}$
- $\mathcal{Y} = \mathbb{R}$
- $p(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y-1+2x)^2}$

Plan

Code (M, n)

$\boxed{\text{Un code } (M, n) \text{ pour le canal } (\mathcal{X}^n, \mathcal{Y}^n, p(\mathbf{y}|\mathbf{x}))}$ est composé de 3 éléments

- Un ensemble de M **messages**. On notera cet ensemble $\mathcal{M} = \{0, 1, \dots, M-1\}$
- Une fonction d'**encodage** (ou encodeur) notée ϕ :

$$\begin{array}{rcl} \phi : \mathcal{M} & \to & \mathcal{X}^n \\ W & \mapsto & \mathbf{X} = \phi(W) \end{array}$$

$$\boxed{\phi(\cdot) \text{ doit être } \textbf{injective}}$$

- Une fonction de **décodage** (ou décodeur) notée ψ :

$$\begin{array}{rcl} \psi : \mathcal{Y}^n & \to & \mathcal{M} \\ \mathbf{Y} & \mapsto & \hat{W} = \psi(\mathbf{Y}) \end{array}$$

$$\boxed{\psi(\cdot) \text{ doit être } \textbf{surjective}}$$

Plan

Probabilité d'erreur

Si le mot de code $W = w$ est envoyé, une erreur se produit ssi $\hat{W} \neq w$.

La probabilité associée à cet événement est notée

$$\begin{aligned} \lambda_w &= \mathbb{P}\left(\hat{W} \neq w | W = w\right) \\ &= \mathbb{P}\left(\psi(\mathbf{Y}) \neq w | W = w\right) \end{aligned}$$

Définitions

- **Probabilité d'erreur maximale** : $P_m^{(n)} = \max_w \lambda_w$
- **Probabilité d'erreur moyenne** : $P_e^{(n)} = \mathbb{P}\left(\hat{W} \neq W\right) = \frac{1}{M} \sum_{w=0}^{M-1} \lambda_w$

Décodage du Maximum a Posteriori (MAP)

Définition

- Soit $\mathcal{C}$ un code (M, n) donné.

- Le **décodeur** du **Maximum A Posteriori** (**MAP**) est la fonction de $\mathbf{y}$ définie par :

$$\Psi_{MAP}(\mathbf{y}) = \underset{w \in \mathcal{M}}{\operatorname{argmax}} \, \mathbb{P}(W = w | \mathbf{Y} = \mathbf{y})$$

Le décodeur MAP minimise P_e

Décodage MAP sur canaux classiques

Soit le **décodeur** MAP défini par : $\Psi_{MAP}(\mathbf{y}) = \underset{w \in \mathcal{M}}{\operatorname{argmax}} \, \mathbb{P}(W = w | \mathbf{Y} = \mathbf{y})$

1. Sur canal BSC : $\Psi_{MAP}(\mathbf{y}) = \underset{\mathbf{x} \in \mathcal{C}}{\operatorname{argmin}} \, d_H(\mathbf{x}, \mathbf{y})$
2. Sur canal AWGN : $\Psi_{MAP}(\mathbf{y}) = \underset{\mathbf{x} \in \mathcal{C}}{\operatorname{argmin}} \, d_E(\mathbf{x}, \mathbf{y})$

Sans structure sur $\mathcal{C}$, ces deux décodeurs sont trop complexes !

Décodage du Maximum a Posteriori (MAP-bit)

Définition

- Soit $\mathcal{C}$ un code **binaire** (k, n) donné.
- Le **décodeur** du **Maximum A Posteriori bit** (**MAP-bit**) est la fonction de **y** définie par :

$$\Psi^{(j)}_{MAP-bit}(\mathbf{y}) = \underset{u \in \{0,1\}}{\operatorname{argmax}} \, \mathbb{P}(U_j = u | \mathbf{Y} = \mathbf{y})$$

- En pratique on calcule les **Logarithmes de rapports de vraisemblances** (LLR) :

$$L(U_i) = \log \frac{\mathbb{P}(U_i = 0|\mathbf{y})}{\mathbb{P}(U_i = 1|\mathbf{y})}$$

- **Le décodeur MAP minimise P_b (la probabilité d'erreur binaire)**
- **Le signe des LLRs : décisions MAP-bit**
- **Le module des LLRs : fiabilité des décisions**

Plan

Enjeux du codage

Compromis entre

- La **taille** du code (n)
- Le **rendement de code** (le débit)
- La **probabilité d'erreur** (maximale ou moyenne)
- La **complexité** de l'encodage
- La **complexité** du décodage

Efficacité spectrale $\Longleftrightarrow$ **Codage** $\Longleftrightarrow$ **Efficacité énergétique**

Plan

Avant de commencer...

Remarques

1. Dans cette section $\mathcal{X} = \mathcal{Y} = \{0, 1\}$ et le canal considéré est le **canal binaire symétrique**
2. Dans cette section on notera $\mathbb{F}_2$ le **corps** $(\{0, 1\}, \oplus, \cdot)$ où :
   - Pour $x, y \in \mathbb{F}_2$, $x \oplus y = (x + y) \mod 2$ ($\equiv$ OU exclusif)
   - Pour $x, y \in \mathbb{F}_2$, $x \cdot y$ est le produit "classique" entre x et y ($\equiv$ ET)
3. $\mathbb{F}_2$ est un corps fini à deux éléments ($\mathbb{Z}/2\mathbb{Z}$)
4. Par la suite on notera $\oplus \rightsquigarrow +$
5. $(\mathbb{F}_2^n, +, \cdot)$ est un **espace vectoriel** où
   - Pour $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$, $\quad \mathbf{x} + \mathbf{y} = [x_0 + y_0, x_1 + y_1 \ldots, x_{n-1} + y_{n-1}]$
   - Pour $x \in \mathbb{F}_2$ et $\mathbf{y} \in \mathbb{F}_2^n$, $\quad x \cdot \mathbf{y} = [x \cdot y_0, x \cdot y_1, \ldots, x \cdot y_{n-1}]$

Code linéaire en bloc

Code linéaire

Soit $\mathcal{C}$ un code $(M = 2^k, n)$, $\mathcal{C}$ est un **code bianire linéaire** si et seulement si les mots de codes $\mathbf{c} \in \mathbb{F}_2^n$ sont obtenus à partir des messages $\mathbf{u} \in \mathbb{F}_2^k$ par la relation

$$\mathbf{c} = \mathbf{u}G$$

où G est une matrice de taille $k \times n$ appelée **matrice génératrice** de $\mathcal{C}$

$$G = \begin{pmatrix} \mathbf{g_0} \\ \mathbf{g_1} \\ \vdots \\ \mathbf{g_{k-1}} \end{pmatrix} = \begin{pmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{pmatrix}$$

Remarques

1. $\mathcal{C}$ est un sous-espace vectoriel de $\mathbb{F}_2^n$ de dimension $rang(G) = k$
2. Il existe plusieurs matrices génératrices pour un même code.
3. le rendement du code est $R = \frac{rang(G)}{n} = \frac{k}{n}$

Plan

Code dual | Matrice de parité

Matrice de parité

Le code $\mathcal{C}$ peut aussi être défini par sa **matrice de parité** H de taille $n-k \times n$:

$$H = \begin{pmatrix} \mathbf{h_0} \\ \mathbf{h_1} \\ \vdots \\ \mathbf{h_{n-k-1}} \end{pmatrix} = \begin{pmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,n-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,n-1} \\ \vdots & \vdots & & \vdots \\ h_{n-k-1,0} & h_{n-k-1,1} & \cdots & h_{n-k-1,n-1} \end{pmatrix}$$

Soit $\mathbf{v} \in \mathbb{F}_2^n$, $\mathbf{v} \in \mathcal{C}$ ($\mathbf{v}$ est un mot de code) si et seulement si

$$\mathbf{v}H^T = 0$$

1. H est appelée **matrice de parité** du code $\mathcal{C}$ et vérifie $GH^T = 0_{k \times n-k}$
2. H n'est pas unique

Codes linéaires en blocs

Définitions

1. À partir de sa **matrice génératrice** G de taille $k \times n$: $\mathcal{C} = \left\{\mathbf{u}G \mid \mathbf{u} \in \mathbb{F}_2^k\right\}$
2. À partir de sa **matrice de parité** H de taille $n - k \times n$: $\mathcal{C} = \left\{\mathbf{c} \in \mathbb{F}_2^n \mid \mathbf{c}H^T = \mathbf{0}\right\}$

1. G et H **ne sont pas uniques**
2. G et H vérifient $GH^T = 0_{k \times n-k}$. Vrai pour tout couple de matrices (G, H) définissant un même code
3. Pour un code binaire : $k \leq n \Rightarrow$ le codage "**ajoute de la redondance**"
4. **Rendement de code** :
$$R = \frac{rang(G)}{n} = \frac{n - rang(H)}{n}$$

Plan

Encodeur systématique

Soit $\mathcal{C}$ un code $(M = 2^k, n)$ pour un canal à entrées binaires. Un encodeur $\varphi(\cdot)$ est dit **systématique** ssi

$$\forall \mathbf{u} \in \mathbb{F}_2^k, \varphi(\mathbf{u}) = [\mathbf{p}\ \mathbf{u}] \text{ avec } \mathbf{p} \in \mathbb{F}_2^{n-k}$$

Si $\mathcal{C}$ est linéaire alors il existe une matrice génératrice sous la forme

$$G = \begin{pmatrix} p_{0,0} & \dots & p_{0,n-k-1} & 1 & 0 & \dots & 0 \\ p_{1,0} & \dots & p_{1,n-k-1} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{k,0} & \dots & p_{k,n-k-1} & 0 & 0 & \dots & 1 \end{pmatrix} = [P \quad I_k]$$

La matrice de parité associée à la matrice G précédente

$$H = \begin{pmatrix} 1 & 0 & \dots & 0 & p_{0,0} & \dots & p_{k,0} \\ 0 & 1 & \dots & 0 & p_{0,1} & \dots & p_{k,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & p_{0,n-k-1} & \dots & p_{k,n-k-1} \end{pmatrix} = [I_{n-k} \quad P^T]$$

Remarques sur les encodeurs systématiques

$$G = \begin{pmatrix} p_{0,0} & \dots & p_{0,n-k-1} & 1 & 0 & \dots & 0 \\ p_{1,0} & \dots & p_{1,n-k-1} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{k,0} & \dots & p_{k,n-k-1} & 0 & 0 & \dots & 1 \end{pmatrix} = [P \quad I_k]$$

1. Un encodeur systématique comporte le message **en clair**

Remarques sur les encodeurs systématiques

$$G = \begin{pmatrix} p_{0,0} & \dots & p_{0,n-k-1} & 1 & 0 & \dots & 0 \\ p_{1,0} & \dots & p_{1,n-k-1} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{k,0} & \dots & p_{k,n-k-1} & 0 & 0 & \dots & 1 \end{pmatrix} = [P \quad I_k]$$

1. Un encodeur systématique comporte le message **en clair**
2. Les encodeurs systématiques sont souvent moins complexes que leurs équivalents non-systématiques

Remarques sur les encodeurs systématiques

$$G = \begin{pmatrix} p_{0,0} & \dots & p_{0,n-k-1} & 1 & 0 & \dots & 0 \\ p_{1,0} & \dots & p_{1,n-k-1} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{k,0} & \dots & p_{k,n-k-1} & 0 & 0 & \dots & 1 \end{pmatrix} = [P \quad I_k]$$

1. Un encodeur systématique comporte le message **en clair**
2. Les encodeurs systématiques sont souvent moins complexes que leurs équivalents non-systématiques
3. Une matrice d'encodage systématique peut être trouvée pour tout code linéaire en bloc de matrice génératrice **pleine** (à des permutations de colonnes près)
   $\rightsquigarrow$ **Pivot de Gauss**

Exemple de Pivot de Gauss

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

1. But : permuter | sommer des lignes pour faire apparaître la matrice I à droite
2. Cette procédure ne donne pas tout le temps une matrice de la forme $G = [P, I]$
3. Si G est de rang plein on peut toujours se ramener à $[P, I]$ **à une permutation de colonne près**
4. Soit $G' = [P, I_k] = G\Pi$ où Π est une matrice de permutation des colonnes, soit $H' = [I_{n-k} P^T]$ alors

$$G'(H')^T = 0_{k \times n-k} = GH^T \text{ avec } H = H'\Pi^T$$

Exemple de Pivot de Gauss

$$G = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \leftarrow \textbf{Pivot}$$

1. But : permuter | sommer des lignes pour faire apparaître la matrice I à droite
2. Cette procédure ne donne pas tout le temps une matrice de la forme $G = [P, I]$
3. Si G est de rang plein on peut toujours se ramener à $[P, I]$ **à une permutation de colonne près**
4. Soit $G' = [P, I_k] = G\Pi$ où Π est une matrice de permutation des colonnes, soit $H' = [I_{n-k} P^T]$ alors

$$G'(H')^T = 0_{k \times n-k} = GH^T \text{ avec } H = H'\Pi^T$$

Exemple de Pivot de Gauss

$$G = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{array}{l} \leftarrow \textbf{Pivot} \\ L_2 \leftarrow L_2 + L_1 \\ \\ \\ \end{array}$$

1. But : permuter | sommer des lignes pour faire apparaître la matrice I à droite
2. Cette procédure ne donne pas tout le temps une matrice de la forme $G = [P, I]$
3. Si G est de rang plein on peut toujours se ramener à $[P, I]$ **à une permutation de colonne près**
4. Soit $G' = [P, I_k] = G\Pi$ où Π est une matrice de permutation des colonnes, soit $H' = [I_{n-k} P^T]$ alors

$$G'(H')^T = 0_{k \times n-k} = GH^T \text{ avec } H = H'\Pi^T$$

Exemple de Pivot de Gauss

$$G = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \leftarrow \textbf{Pivot}$$

1. But : permuter | sommer des lignes pour faire apparaître la matrice I à droite
2. Cette procédure ne donne pas tout le temps une matrice de la forme $G = [P, I]$
3. Si G est de rang plein on peut toujours se ramener à $[P, I]$ **à une permutation de colonne près**
4. Soit $G' = [P, I_k] = G\Pi$ où Π est une matrice de permutation des colonnes, soit $H' = [I_{n-k} P^T]$ alors

$$G'(H')^T = 0_{k\times n-k} = GH^T \text{ avec } H = H'\Pi^T$$

Exemple de Pivot de Gauss

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{array}{l} \\ \leftarrow \textbf{Pivot} \\ L_3 \leftarrow L_3 + L_2 \\ \\ \end{array}$$

1. But : permuter | sommer des lignes pour faire apparaître la matrice I à droite
2. Cette procédure ne donne pas tout le temps une matrice de la forme $G = [P, I]$
3. Si G est de rang plein on peut toujours se ramener à $[P, I]$ **à une permutation de colonne près**
4. Soit $G' = [P, I_k] = G\Pi$ où Π est une matrice de permutation des colonnes, soit $H' = [I_{n-k} P^T]$ alors

$$G'(H')^T = 0_{k\times n-k} = GH^T \text{ avec } H = H'\Pi^T$$

Exemple de Pivot de Gauss

$$G = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{matrix} \\ \\ \leftarrow \textbf{Pivot} \\ \\ \end{matrix}$$

1. But : permuter | sommer des lignes pour faire apparaître la matrice I à droite
2. Cette procédure ne donne pas tout le temps une matrice de la forme $G = [P, I]$
3. Si G est de rang plein on peut toujours se ramener à $[P, I]$ **à une permutation de colonne près**
4. Soit $G' = [P, I_k] = G\Pi$ où Π est une matrice de permutation des colonnes, soit $H' = [I_{n-k} P^T]$ alors

$$G'(H')^T = 0_{k\times n-k} = GH^T \text{ avec } H = H'\Pi^T$$

Exemple de Pivot de Gauss

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{array}{l} \\ \\ \leftarrow \textbf{Pivot} \\ L_4 \leftarrow L_4 + L_3 \end{array}$$

1. But : permuter | sommer des lignes pour faire apparaître la matrice I à droite
2. Cette procédure ne donne pas tout le temps une matrice de la forme $G = [P, I]$
3. Si G est de rang plein on peut toujours se ramener à $[P, I]$ **à une permutation de colonne près**
4. Soit $G' = [P, I_k] = G\Pi$ où Π est une matrice de permutation des colonnes, soit $H' = [I_{n-k} P^T]$ alors

$$G'(H')^T = 0_{k \times n-k} = GH^T \text{ avec } H = H'\Pi^T$$

Plan

Décodage MAP-bit

Décodage MAP-bit

- Le **décodeur MAP-bit** encodage systématique :

$$\Psi^{(j)}_{MAP-bit}(\mathbf{y}) = \underset{x_j \in \{0,1\}}{\operatorname{argmax}} \, \mathbb{P}(X_j = x_j | \mathbf{Y} = \mathbf{y})$$

Décodage MAP-bit

Décodage MAP-bit

- Le **décodeur MAP-bit** encodage systématique :

$$\Psi^{(j)}_{MAP-bit}(\mathbf{y}) = \underset{x_j \in \{0,1\}}{\text{argmax}}\, \mathbb{P}(X_j = x_j | \mathbf{Y} = \mathbf{y})$$

- Le **décodeur MAP-bit** encodage systématique (2) :

$$\Psi^{(j)}_{MAP-bit}(\mathbf{y}) = \underset{x_j \in \{0,1\}}{\text{argmax}} \sum_{\mathbf{x}_{\sim j} \in \mathbb{F}_2^{n-1}} \mathbb{P}(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}) \mathbb{1}(\mathbf{x}H^T = \mathbf{0})$$

Décodage MAP-bit

Décodage MAP-bit

- Le **décodeur MAP-bit** encodage systématique :

$$\Psi^{(j)}_{MAP-bit}(\mathbf{y}) = \underset{x_j\in\{0,1\}}{\text{argmax}}\, \mathbb{P}(X_j = x_j|\mathbf{Y} = \mathbf{y})$$

- Le **décodeur MAP-bit** encodage systématique (2) :

$$\begin{aligned}\Psi^{(j)}_{MAP-bit}(\mathbf{y}) &= \underset{x_j\in\{0,1\}}{\text{argmax}} \sum_{\mathbf{x}_{\sim j}\in\mathbb{F}_2^{n-1}} \mathbb{P}(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x})\mathbb{1}(\mathbf{x}H^T = \mathbf{0}) \\ &= \underset{x_j\in\{0,1\}}{\text{argmax}} \sum_{\mathbf{x}_{\sim j}\in\mathbb{F}_2^{n-1}} \prod_{i=0}^{n-1} \mathbb{P}(Y_i = y_i|X_i = x_i)\mathbb{1}(\mathbf{x}H^T = \mathbf{0})\end{aligned}$$

Décodage MAP-bit

Décodage MAP-bit

- Le **décodeur MAP-bit** encodage systématique :

$$\Psi^{(j)}_{MAP-bit}(\mathbf{y}) = \underset{x_j \in \{0,1\}}{\operatorname{argmax}} \mathbb{P}(X_j = x_j | \mathbf{Y} = \mathbf{y})$$

- Le **décodeur MAP-bit** encodage systématique (2) :

$$\begin{aligned}
\Psi^{(j)}_{MAP-bit}(\mathbf{y}) &= \underset{x_j \in \{0,1\}}{\operatorname{argmax}} \sum_{\mathbf{x}_{\sim j} \in \mathbb{F}_2^{n-1}} \mathbb{P}(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}) \mathbb{1}(\mathbf{x}H^T = \mathbf{0}) \\
&= \underset{x_j \in \{0,1\}}{\operatorname{argmax}} \sum_{\mathbf{x}_{\sim j} \in \mathbb{F}_2^{n-1}} \prod_{i=0}^{n-1} \mathbb{P}(Y_i = y_i | X_i = x_i) \mathbb{1}(\mathbf{x}H^T = \mathbf{0})
\end{aligned}$$

Sans structure sur $\mathcal{C}$, ce décodeur est aussi trop complexe !

Plan

Plan

Codes Low Density Parity Check (LDPC)

- Introduits par Gallager pendant sa **thèse de doctorat** en 1963

Codes Low Density Parity Check (LDPC)

- Introduits par Gallager pendant sa **thèse de doctorat** en 1963
  → Codes possédant une **matrice de parité peu dense**

Codes Low Density Parity Check (LDPC)

- Introduits par Gallager pendant sa **thèse de doctorat** en 1963
  - → Codes possédant une **matrice de parité peu dense**
  - → Codes pouvant être analysés (exposant d'erreur)

Codes Low Density Parity Check (LDPC)

- Introduits par Gallager pendant sa **thèse de doctorat** en 1963
  - → Codes possédant une **matrice de parité peu dense**
  - → Codes pouvant être analysés (exposant d'erreur)
  - → Décodage simplifié

Codes Low Density Parity Check (LDPC)

- Introduits par Gallager pendant sa **thèse de doctorat** en 1963
  - $\rightarrow$ Codes possédant une **matrice de parité peu dense**
  - $\rightarrow$ Codes pouvant être analysés (exposant d'erreur)
  - $\rightarrow$ Décodage simplifié
- **Peu de travaux** pendant $\sim$ 30 ans (Tanner en 1981)

Codes Low Density Parity Check (LDPC)

- Introduits par Gallager pendant sa **thèse de doctorat** en 1963
  - → Codes possédant une **matrice de parité peu dense**
  - → Codes pouvant être analysés (exposant d'erreur)
  - → Décodage simplifié
- **Peu de travaux** pendant $\sim$ 30 ans (Tanner en 1981)
  - → Codes représentable à l'aide d'un **graphe bipartite** (**graphe de Tanner**)

Codes Low Density Parity Check (LDPC)

- Introduits par Gallager pendant sa **thèse de doctorat** en 1963
  - $\rightarrow$ Codes possédant une **matrice de parité peu dense**
  - $\rightarrow$ Codes pouvant être analysés (exposant d'erreur)
  - $\rightarrow$ Décodage simplifié

- **Peu de travaux** pendant $\sim$ 30 ans (Tanner en 1981)
  - $\rightarrow$ Codes représentable à l'aide d'un **graphe bipartite** (**graphe de Tanner**)
  - $\rightarrow$ Décodage possible à l'aide du graphe

Codes Low Density Parity Check (LDPC)

- Introduits par Gallager pendant sa **thèse de doctorat** en 1963
  - $\rightarrow$ Codes possédant une **matrice de parité peu dense**
  - $\rightarrow$ Codes pouvant être analysés (exposant d'erreur)
  - $\rightarrow$ Décodage simplifié
- **Peu de travaux** pendant $\sim$ 30 ans (Tanner en 1981)
  - $\rightarrow$ Codes représentable à l'aide d'un **graphe bipartite** (**graphe de Tanner**)
  - $\rightarrow$ Décodage possible à l'aide du graphe
  - $\rightarrow$ Performances dépendant des propriétés du graphe

Codes Low Density Parity Check (LDPC)

- Introduits par Gallager pendant sa **thèse de doctorat** en 1963
  - → Codes possédant une **matrice de parité peu dense**
  - → Codes pouvant être analysés (exposant d'erreur)
  - → Décodage simplifié

- **Peu de travaux** pendant $\sim$ 30 ans (Tanner en 1981)
  - → Codes représentable à l'aide d'un **graphe bipartite** (**graphe de Tanner**)
  - → Décodage possible à l'aide du graphe
  - → Performances dépendant des propriétés du graphe

- Algorithme de propagation de croyance (IA) (Pearl en 1988)

Codes Low Density Parity Check (LDPC)

- Introduits par Gallager pendant sa **thèse de doctorat** en 1963
  - → Codes possédant une **matrice de parité peu dense**
  - → Codes pouvant être analysés (exposant d'erreur)
  - → Décodage simplifié

- **Peu de travaux** pendant $\sim$ 30 ans (Tanner en 1981)
  - → Codes représentable à l'aide d'un **graphe bipartite** (**graphe de Tanner**)
  - → Décodage possible à l'aide du graphe
  - → Performances dépendant des propriétés du graphe

- Algorithme de propagation de croyance (IA) (Pearl en 1988)
  - → Alogrithme de propagation de croyance (BP - Belief Propagation)

Codes Low Density Parity Check (LDPC)

- Introduits par Gallager pendant sa **thèse de doctorat** en 1963
  - → Codes possédant une **matrice de parité peu dense**
  - → Codes pouvant être analysés (exposant d'erreur)
  - → Décodage simplifié
- **Peu de travaux** pendant $\sim$ 30 ans (Tanner en 1981)
  - → Codes représentable à l'aide d'un **graphe bipartite** (**graphe de Tanner**)
  - → Décodage possible à l'aide du graphe
  - → Performances dépendant des propriétés du graphe
- Algorithme de propagation de croyance (IA) (Pearl en 1988)
  - → Alogrithme de propagation de croyance (BP - Belief Propagation)
- **Redécouverte** des codes LDPC (MacKay, Luby fin 1990)

Codes Low Density Parity Check (LDPC)

- Introduits par Gallager pendant sa **thèse de doctorat** en 1963
  - → Codes possédant une **matrice de parité peu dense**
  - → Codes pouvant être analysés (exposant d'erreur)
  - → Décodage simplifié
- **Peu de travaux** pendant $\sim$ 30 ans (Tanner en 1981)
  - → Codes représentable à l'aide d'un **graphe bipartite** (**graphe de Tanner**)
  - → Décodage possible à l'aide du graphe
  - → Performances dépendant des propriétés du graphe
- Algorithme de propagation de croyance (IA) (Pearl en 1988)
  - → Alogrithme de propagation de croyance (BP - Belief Propagation)
- **Redécouverte** des codes LDPC (MacKay, Luby fin 1990)
  - → (Re)Montrent que les codes LDPC sont de bons codes

Codes Low Density Parity Check (LDPC)

- Introduits par Gallager pendant sa **thèse de doctorat** en 1963
  - $\rightarrow$ Codes possédant une **matrice de parité peu dense**
  - $\rightarrow$ Codes pouvant être analysés (exposant d'erreur)
  - $\rightarrow$ Décodage simplifié
- **Peu de travaux** pendant $\sim$ 30 ans (Tanner en 1981)
  - $\rightarrow$ Codes représentable à l'aide d'un **graphe bipartite** (**graphe de Tanner**)
  - $\rightarrow$ Décodage possible à l'aide du graphe
  - $\rightarrow$ Performances dépendant des propriétés du graphe
- Algorithme de propagation de croyance (IA) (Pearl en 1988)
  - $\rightarrow$ Alogrithme de propagation de croyance (BP - Belief Propagation)
- **Redécouverte** des codes LDPC (MacKay, Luby fin 1990)
  - $\rightarrow$ (Re)Montrent que les codes LDPC sont de bons codes

Plan

Définition des codes LDPC

Définitions

1. Soit une matrice H

$$H = \begin{pmatrix} h_{0,0} & h_{0,1} & \dots & h_{0,n-1} \\ h_{1,0} & h_{1,1} & \dots & h_{1,n-1} \\ \vdots & \vdots & & \vdots \\ h_{m-1,0} & h_{m-1,1} & \dots & h_{m-1,n-1} \end{pmatrix}$$

   Densité de H : $\dfrac{\left|\{i,j : h_{i,j} = 1\}\right|}{m\, n}$

2. **Codes LDPC** : Codes possédant une matrice de parité H peu dense (creuse). Ordre de grandeur pour n grand ≤ 0.01.
3. **Codes réguliers** : poids des lignes constant r, poids des colonnes constant g
4. Rendement d'un code LDPC régulier : $R \geq 1 - \dfrac{m}{n} = 1 - \dfrac{g}{r}$
5. $R_d = 1 - \dfrac{g}{r}$ est appelé **rendement de construction** d'un code LDPC

Définition des codes LDPC

Petit TD dans le cours...

Soit une matrice H

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

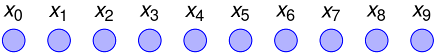
1. Donner la densité de H
2. H définie-t-elle un code LDPC régulier ?
3. Si oui, que valent g et r ?
4. Combien vaut le rendement de construction de ce code ?
5. Combien vaut le rendement de ce code ?

Plan

Graphe de Tanner

Le **graphe de Tanner** est un **graphe bipartite** avec :

1. n **nœuds de variables** représentant les variables v_j $j \in \{0, \ldots n-1\}$
2. m **nœuds de parité** p_i $i \in \{0, \ldots m-1\}$
3. Une arrête est dessinée entre nœud de variable x_j et le nœud de parité c_i ssi $h_{i,j} = 1$



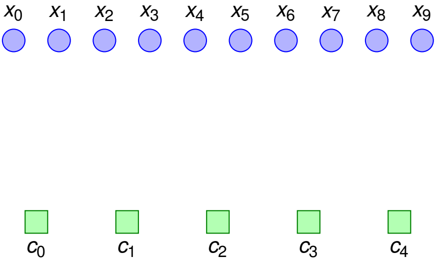
$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Graphe de Tanner

Le **graphe de Tanner** est un **graphe bipartite** avec :

1. n **nœuds de variables** représentant les variables v_j $j \in \{0, \dots n-1\}$
2. m **nœuds de parité** p_i $i \in \{0, \dots m-1\}$
3. Une arrête est dessinée entre nœud de variable x_j et le nœud de parité c_i ssi $h_{i,j} = 1$

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

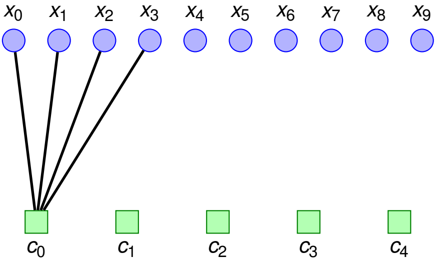


Graphe de Tanner

Le **graphe de Tanner** est un **graphe bipartite** avec :

1. n **nœuds de variables** représentant les variables v_j $j \in \{0, \dots n-1\}$
2. m **nœuds de parité** p_i $i \in \{0, \dots m-1\}$
3. Une arrête est dessinée entre nœud de variable x_j et le nœud de parité c_i ssi $h_{i,j} = 1$

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

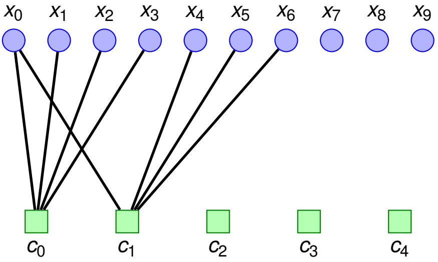


Graphe de Tanner

Le **graphe de Tanner** est un **graphe bipartite** avec :

1. n **nœuds de variables** représentant les variables v_j $j \in \{0, \ldots n-1\}$
2. m **nœuds de parité** p_i $i \in \{0, \ldots m-1\}$
3. Une arrête est dessinée entre nœud de variable x_j et le nœud de parité c_i ssi $h_{i,j} = 1$

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

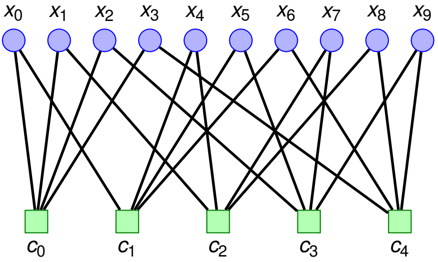


Graphe de Tanner

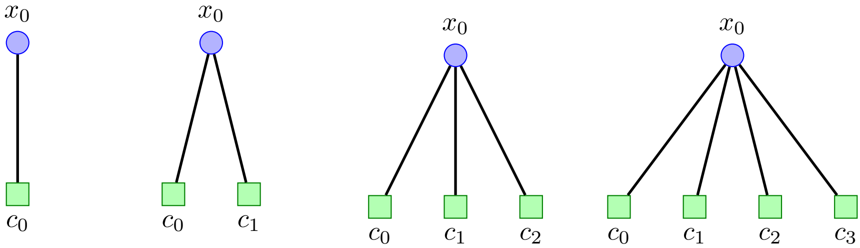
Le **graphe de Tanner** est un **graphe bipartite** avec :

1. n **nœuds de variables** représentant les variables v_j $j \in \{0, \dots n-1\}$
2. m **nœuds de parité** p_i $i \in \{0, \dots m-1\}$
3. Une arrête est dessinée entre nœud de variable x_j et le nœud de parité c_i ssi $h_{i,j} = 1$

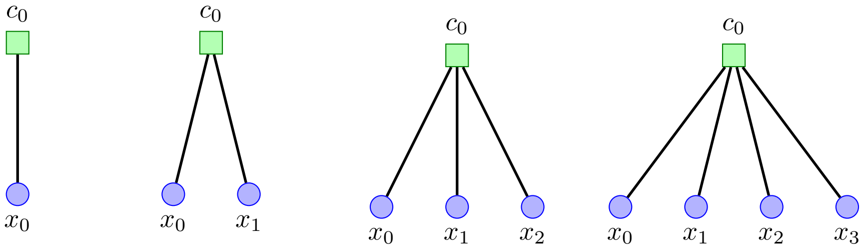
$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$



Degrés des nœuds de variable

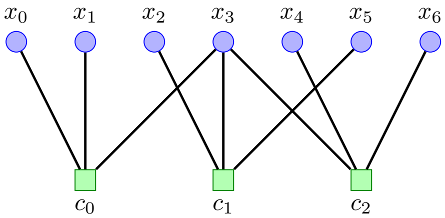


Degrés des nœuds de parité



Codes LDPC irréguliers

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$



Polynôme de distribution des degrés des nœuds de variables : $\lambda(X) = \sum_{d=1}^{d_v} \lambda_d X^{d-1}$

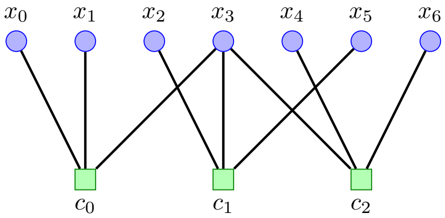
Polynôme de distribution des degrés des nœuds de parités : $\rho(X) = \sum_{d=1}^{d_c} \rho_d X^{d-1}$

Borne sur le rendement du code : $R \geq 1 - \dfrac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}$

Plan

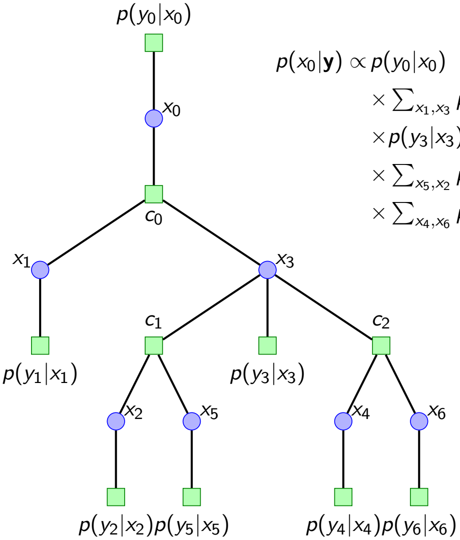
Retour sur le décodage du MAP-bit

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$



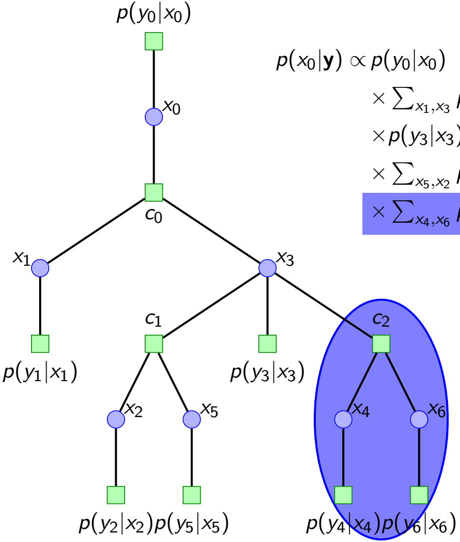
$$\begin{aligned} p(x_0|\mathbf{y}) &\propto \sum_{\mathbf{x}_{\sim 0}} \prod_{i=0}^{6} p(y_i|x_i) \mathbb{1}(\mathbf{x}H^T = \mathbf{0}) \\ &= p(y_0|x_0) \sum_{x_1, x_3} p(y_1|x_1) p(y_3|x_3) \mathbb{1}(x_0 + x_1 + x_3 = 0) \\ &\qquad \times \sum_{x_5, x_2} p(y_2|x_2) p(y_5|x_5) \mathbb{1}(x_2 + x_3 + x_5 = 0) \\ &\qquad \times \sum_{x_4, x_6} p(y_4|x_4) p(y_6|x_6) \mathbb{1}(x_3 + x_4 + x_6 = 0) \end{aligned}$$

Algorithme somme-produit



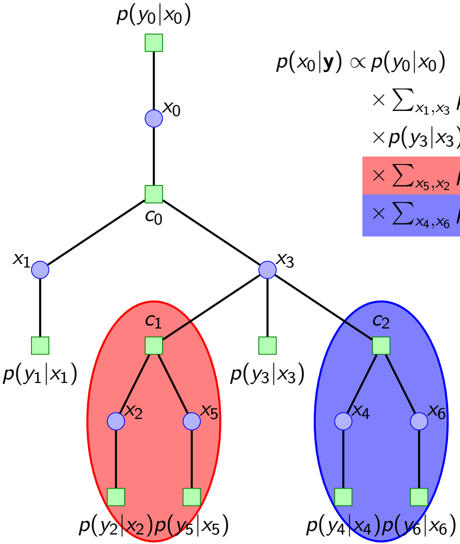
$$\begin{aligned} p(x_0|\mathbf{y}) \propto p(y_0|x_0) & \\ & \times \sum_{x_1,x_3} p(y_1|x_1)\mathbb{1}(x_0+x_1+x_3=0) \\ & \times p(y_3|x_3) \\ & \times \sum_{x_5,x_2} p(y_2|x_2)p(y_5|x_5)\mathbb{1}(x_2+x_3+x_5=0) \\ & \times \sum_{x_4,x_6} p(y_4|x_4)p(y_6|x_6)\mathbb{1}(x_3+x_4+x_6=0) \end{aligned}$$

Algorithme somme-produit



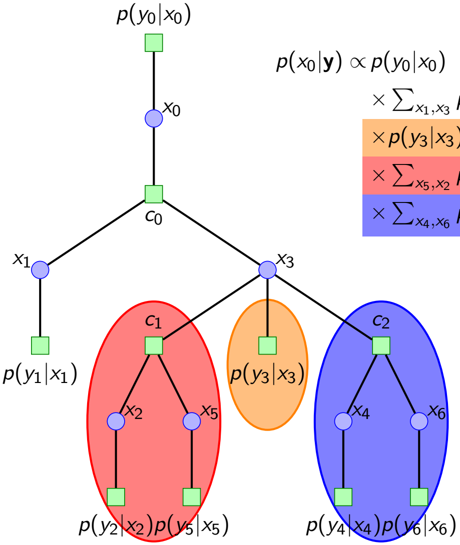
$$\begin{aligned}
p(x_0|\mathbf{y}) \propto\ & p(y_0|x_0) \\
& \times \textstyle\sum_{x_1,x_3} p(y_1|x_1)\mathbb{1}(x_0+x_1+x_3=0) \\
& \times p(y_3|x_3) \\
& \times \textstyle\sum_{x_5,x_2} p(y_2|x_2)p(y_5|x_5)\mathbb{1}(x_2+x_3+x_5=0) \\
& \times \textstyle\sum_{x_4,x_6} p(y_4|x_4)p(y_6|x_6)\mathbb{1}(x_3+x_4+x_6=0)
\end{aligned}$$

Algorithme somme-produit



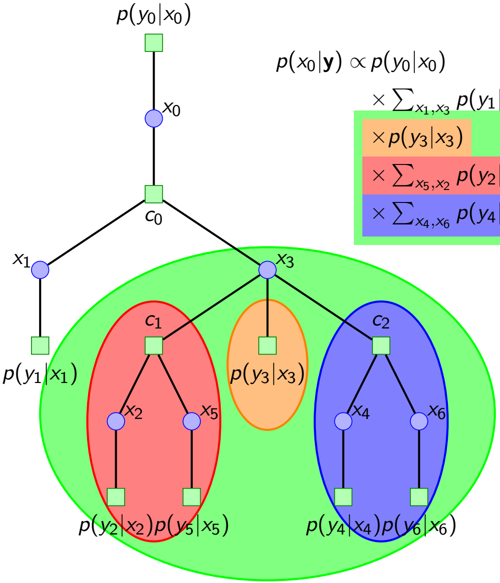
$$
\begin{aligned}
p(x_0|\mathbf{y}) \propto\ & p(y_0|x_0) \\
& \times \sum_{x_1,x_3} p(y_1|x_1)\mathbb{1}(x_0 + x_1 + x_3 = 0) \\
& \times p(y_3|x_3) \\
& \times \sum_{x_5,x_2} p(y_2|x_2)p(y_5|x_5)\mathbb{1}(x_2 + x_3 + x_5 = 0) \\
& \times \sum_{x_4,x_6} p(y_4|x_4)p(y_6|x_6)\mathbb{1}(x_3 + x_4 + x_6 = 0)
\end{aligned}
$$

Algorithme somme-produit



$$
\begin{aligned}
p(x_0|\mathbf{y}) \propto\ & p(y_0|x_0) \\
& \times \textstyle\sum_{x_1,x_3} p(y_1|x_1)\mathbb{1}(x_0+x_1+x_3=0) \\
& \times p(y_3|x_3) \\
& \times \textstyle\sum_{x_5,x_2} p(y_2|x_2)p(y_5|x_5)\mathbb{1}(x_2+x_3+x_5=0) \\
& \times \textstyle\sum_{x_4,x_6} p(y_4|x_4)p(y_6|x_6)\mathbb{1}(x_3+x_4+x_6=0)
\end{aligned}
$$

Algorithme somme-produit



$$p(x_0|\mathbf{y}) \propto p(y_0|x_0)$$

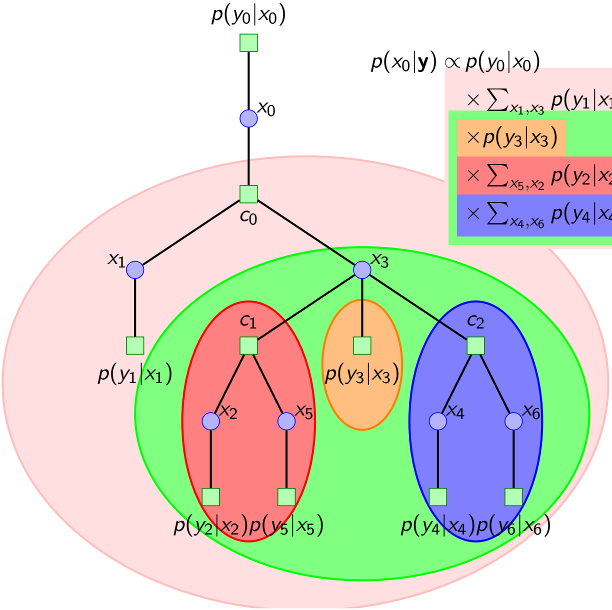
$$\times \sum_{x_1,x_3} p(y_1|x_1)\mathbb{1}(x_0+x_1+x_3=0)$$

$$\times p(y_3|x_3)$$

$$\times \sum_{x_5,x_2} p(y_2|x_2)p(y_5|x_5)\mathbb{1}(x_2+x_3+x_5=0)$$

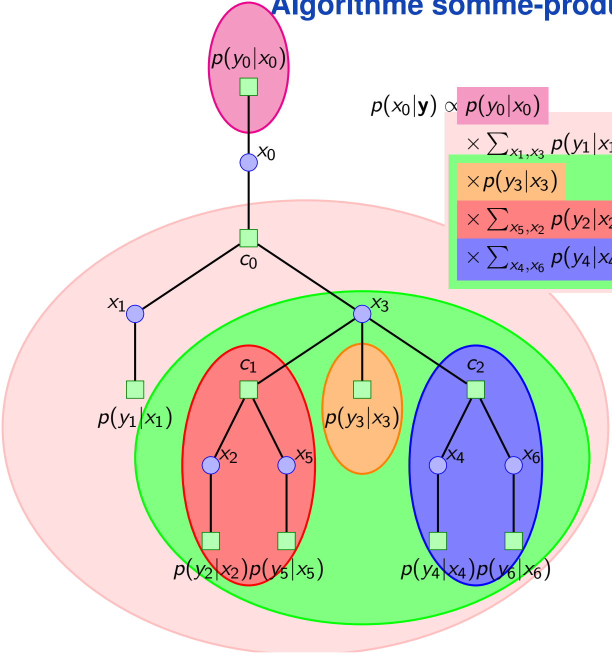
$$\times \sum_{x_4,x_6} p(y_4|x_4)p(y_6|x_6)\mathbb{1}(x_3+x_4+x_6=0)$$

Algorithme somme-produit



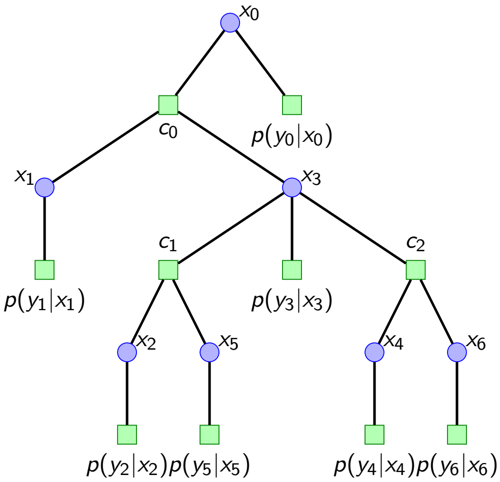
$$
\begin{aligned}
p(x_0|\mathbf{y}) \propto\ & p(y_0|x_0) \\
& \times \sum_{x_1,x_3} p(y_1|x_1)\mathbb{1}(x_0+x_1+x_3=0) \\
& \times p(y_3|x_3) \\
& \times \sum_{x_5,x_2} p(y_2|x_2)p(y_5|x_5)\mathbb{1}(x_2+x_3+x_5=0) \\
& \times \sum_{x_4,x_6} p(y_4|x_4)p(y_6|x_6)\mathbb{1}(x_3+x_4+x_6=0)
\end{aligned}
$$

Algorithme somme-produit

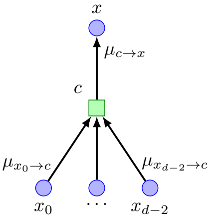


$$
\begin{aligned}
p(x_0|\mathbf{y}) \propto\ & p(y_0|x_0) \\
& \times \sum_{x_1,x_3} p(y_1|x_1)\mathbb{1}(x_0+x_1+x_3=0) \\
& \times p(y_3|x_3) \\
& \times \sum_{x_5,x_2} p(y_2|x_2)p(y_5|x_5)\mathbb{1}(x_2+x_3+x_5=0) \\
& \times \sum_{x_4,x_6} p(y_4|x_4)p(y_6|x_6)\mathbb{1}(x_3+x_4+x_6=0)
\end{aligned}
$$

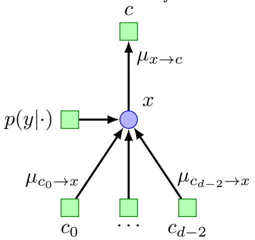
Algorithme somme-produit



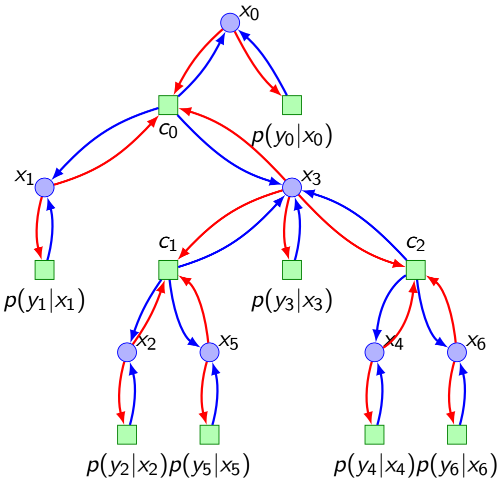
$$\mu_{c\to x}(x) = \sum_{\sim x} \mathbb{1}(x + x_0 + \ldots + x_{d-2} = 0)\times$$
$$\times \prod_0^{d-2} \mu_{x_k \to c}(x_k)$$



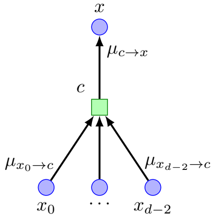
$$\mu_{x\to c}(x) = p(y|x) \prod_{f=0}^{d-2} \mu_{f\to x}(x)$$



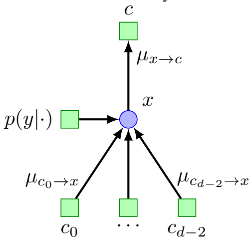
Algorithme somme-produit



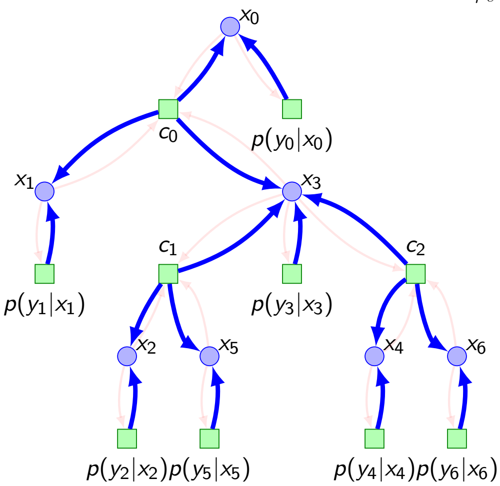
$$\mu_{c\to x}(x) = \sum_{\sim x} \mathbb{1}(x + x_0 + \ldots + x_{d-2} = 0)\times$$
$$\times \prod_0^{d-2} \mu_{x_k\to c}(x_k)$$



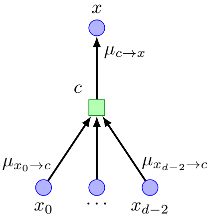
$$\mu_{x\to c}(x) = p(y|x) \prod_{f=0}^{d-2} \mu_{f\to x}(x)$$



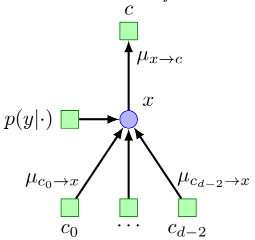
Algorithme somme-produit



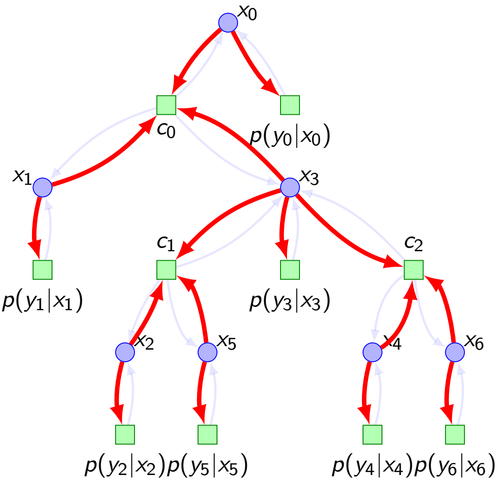
$$\mu_{c\to x}(x) = \sum_{\sim x} \mathbb{1}(x + x_0 + \ldots + x_{d-2} = 0)\times$$
$$\times \prod_0^{d-2} \mu_{x_k\to c}(x_k)$$



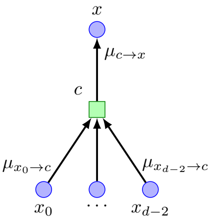
$$\mu_{x\to c}(x) = p(y|x)\prod_{f=0}^{d-2}\mu_{f\to x}(x)$$



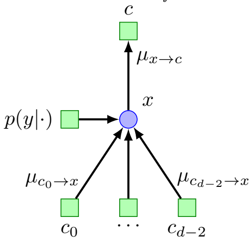
Algorithme somme-produit



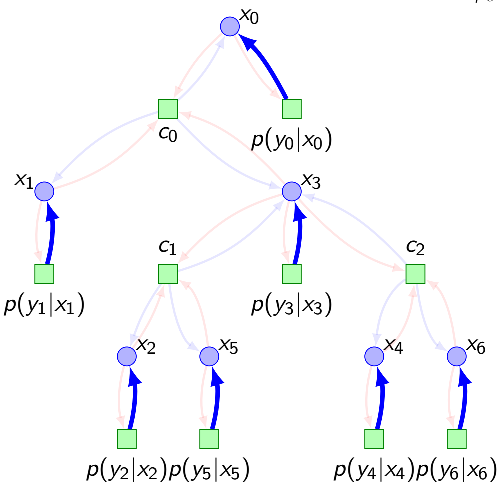
$$\mu_{c\to x}(x) = \sum_{\sim x} \mathbb{1}(x + x_0 + \ldots + x_{d-2} = 0)\times$$
$$\times \prod_{0}^{d-2} \mu_{x_k \to c}(x_k)$$



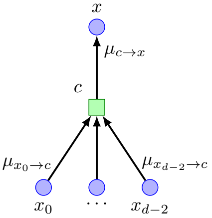
$$\mu_{x\to c}(x) = p(y|x) \prod_{f=0}^{d-2} \mu_{f\to x}(x)$$



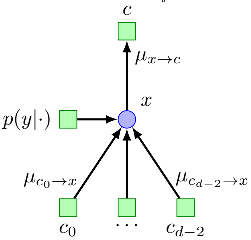
Algorithme somme-produit



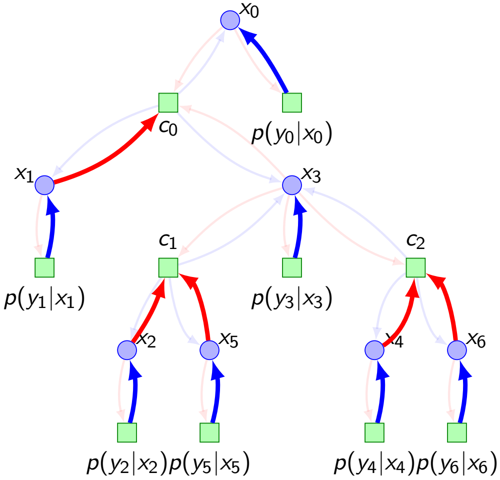
$$\mu_{c\to x}(x) = \sum_{\sim x} \mathbb{1}(x + x_0 + \ldots + x_{d-2} = 0)\times$$
$$\times \prod_0^{d-2} \mu_{x_k\to c}(x_k)$$



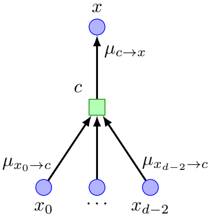
$$\mu_{x\to c}(x) = p(y|x)\prod_{f=0}^{d-2} \mu_{f\to x}(x)$$



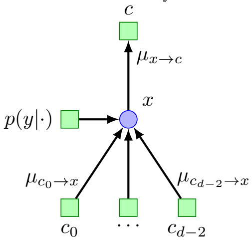
Algorithme somme-produit



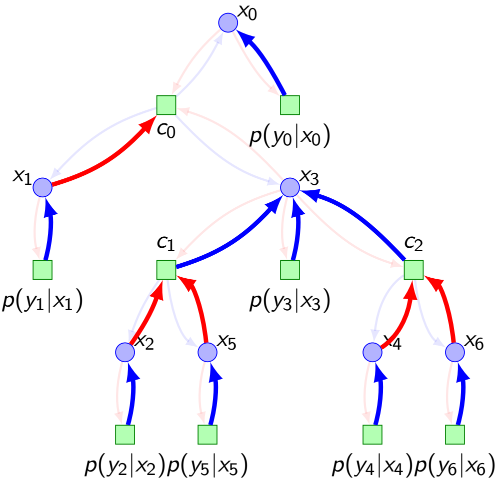
$$\mu_{c\to x}(x) = \sum_{\sim x} \mathbb{1}(x + x_0 + \ldots + x_{d-2} = 0) \times \\ \times \prod_0^{d-2} \mu_{x_k \to c}(x_k)$$



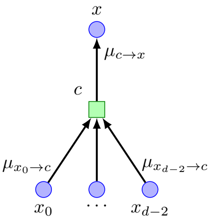
$$\mu_{x\to c}(x) = p(y|x) \prod_{f=0}^{d-2} \mu_{f\to x}(x)$$



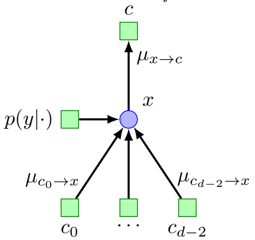
Algorithme somme-produit



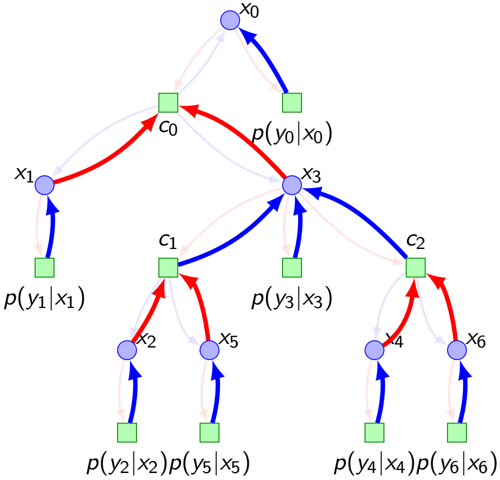
$$\mu_{c\to x}(x) = \sum_{\sim x} \mathbb{1}(x + x_0 + \ldots + x_{d-2} = 0)\times$$
$$\times \prod_0^{d-2} \mu_{x_k\to c}(x_k)$$



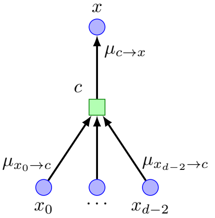
$$\mu_{x\to c}(x) = p(y|x) \prod_{f=0}^{d-2} \mu_{f\to x}(x)$$



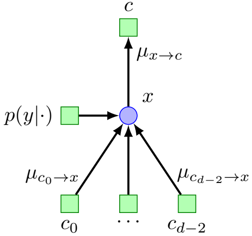
Algorithme somme-produit



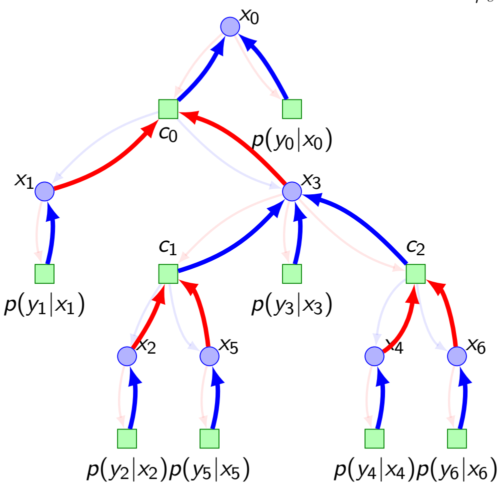
$$\mu_{c\to x}(x) = \sum_{\sim x} \mathbb{1}(x + x_0 + \ldots + x_{d-2} = 0)\times$$
$$\times \prod_0^{d-2} \mu_{x_k\to c}(x_k)$$



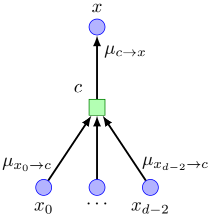
$$\mu_{x\to c}(x) = p(y|x)\prod_{f=0}^{d-2}\mu_{f\to x}(x)$$



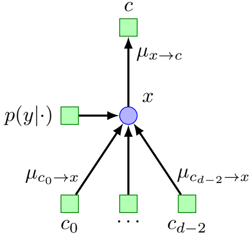
Algorithme somme-produit



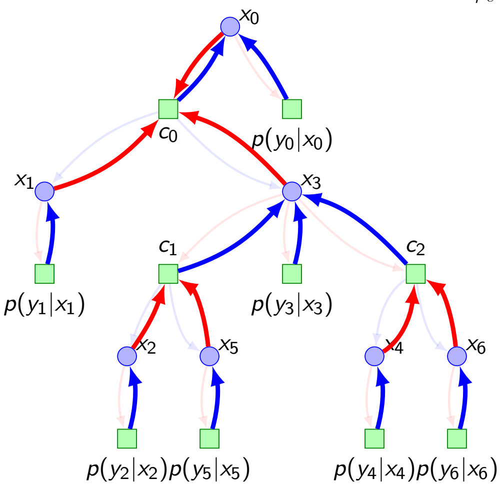
$$\mu_{c\to x}(x) = \sum_{\sim x} \mathbb{1}(x + x_0 + \ldots + x_{d-2} = 0)\times$$
$$\times \prod_0^{d-2} \mu_{x_k\to c}(x_k)$$



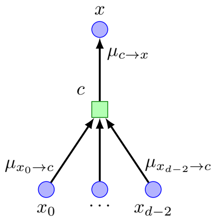
$$\mu_{x\to c}(x) = p(y|x) \prod_{f=0}^{d-2} \mu_{f\to x}(x)$$



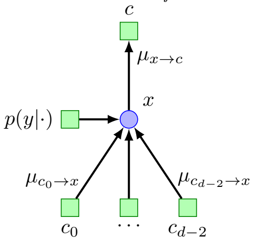
Algorithme somme-produit



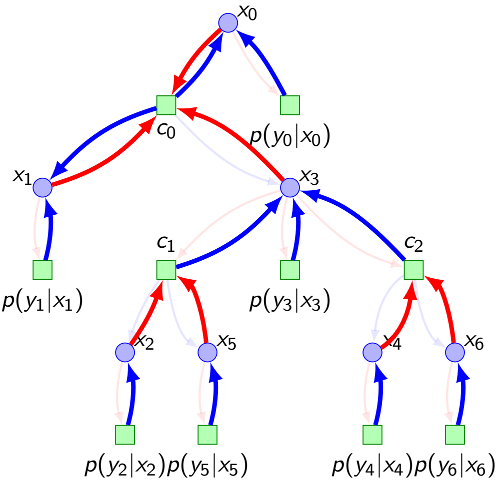
$$\mu_{c\to x}(x) = \sum_{\sim x} \mathbb{1}(x + x_0 + \ldots + x_{d-2} = 0)\times$$
$$\times \prod_0^{d-2} \mu_{x_k\to c}(x_k)$$



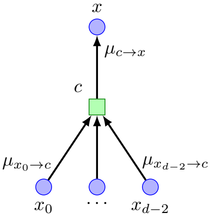
$$\mu_{x\to c}(x) = p(y|x)\prod_{f=0}^{d-2}\mu_{f\to x}(x)$$



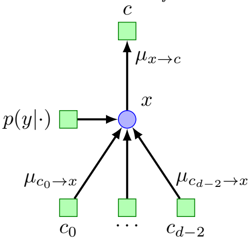
Algorithme somme-produit



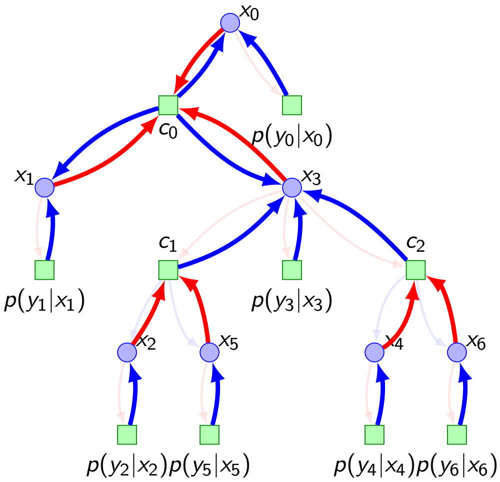
$$\mu_{c\to x}(x) = \sum_{\sim x} \mathbb{1}(x + x_0 + \ldots + x_{d-2} = 0)\times$$
$$\times \prod_0^{d-2} \mu_{x_k\to c}(x_k)$$



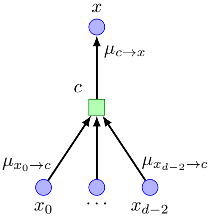
$$\mu_{x\to c}(x) = p(y|x)\prod_{f=0}^{d-2} \mu_{f\to x}(x)$$



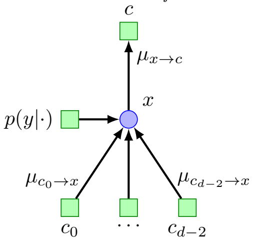
Algorithme somme-produit



$$\mu_{c\to x}(x) = \sum_{\sim x} \mathbb{1}(x + x_0 + \ldots + x_{d-2} = 0)\times$$
$$\times \prod_0^{d-2} \mu_{x_k\to c}(x_k)$$

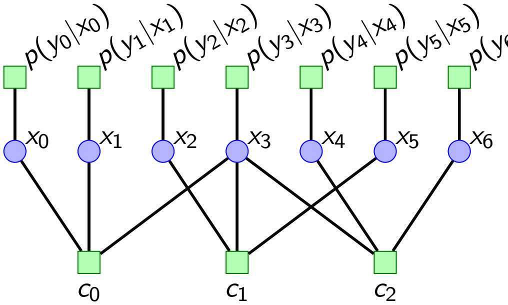


$$\mu_{x\to c}(x) = p(y|x)\prod_{f=0}^{d-2}\mu_{f\to x}(x)$$



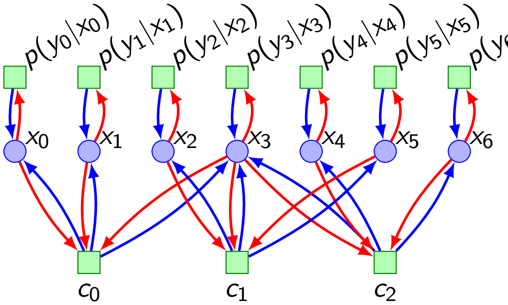
Ordonnancement : flooding

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$



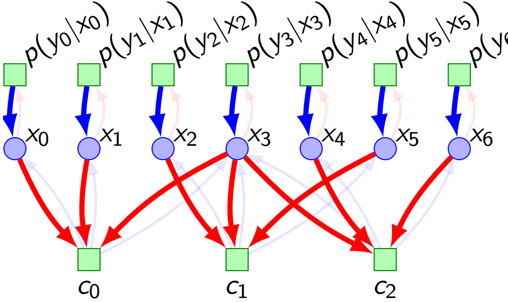
Ordonnancement : flooding

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$



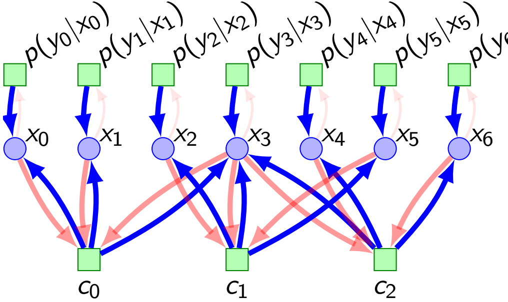
Ordonnancement : flooding

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$



Ordonnancement : flooding

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$



Conclusion sur l'algorithme somme-produit

L'algorithme SP :

- utilise des calcul locaux $\Rightarrow$ réduit complexité

Conclusion sur l'algorithme somme-produit

L'algorithme SP :

- utilise des calcul locaux $\Rightarrow$ réduit complexité
- permet d'approximer les probabilités a posteriori $p(x_i|\mathbf{y})$

Conclusion sur l'algorithme somme-produit

L'algorithme SP :

- utilise des calcul locaux $\Rightarrow$ réduit complexité
- permet d'approximer les probabilités a posteriori $p(x_i|\mathbf{y})$
- ce calcul est **exact** si le **graphe de Tanner est un arbre**

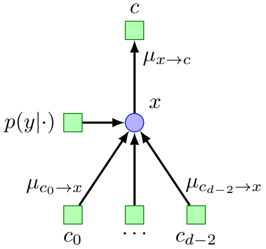
Conclusion sur l'algorithme somme-produit

L'algorithme SP :

- utilise des calcul locaux $\Rightarrow$ réduit complexité
- permet d'approximer les probabilités a posteriori $p(x_i|\mathbf{y})$
- ce calcul est **exact** si le **graphe de Tanner est un arbre**
- si le graphe possède des cycles $\Rightarrow$ itérer

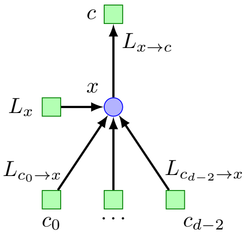
Calculer avec des LLR - Nœuds de variables

$$\mu_{x\to c}(x) = p(y|x)\prod_{f=0}^{d-2}\mu_{f\to x}(x)$$



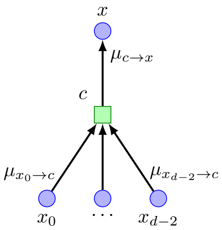
$\Rightarrow$

$$L_{x\to c} = L_x + \sum_{k=0}^{d-2} L_{c_k\to x}$$



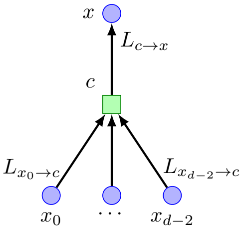
Calculer avec des LLR - Nœuds de variables

$$\mu_{c\to x}(x) = \sum_{\sim x} \mathbb{1}(x + x_0 + \ldots + x_{d-2} = 0)\times \\ \times \prod_0^{d-2} \mu_{x_k\to c}(x_k)$$



$\Rightarrow$

$$L_{c\to x} = 2\,\mathrm{atanh}\left(\prod_{k=0}^{d-2} \tanh \frac{L_{x_k\to c}}{2}\right)$$



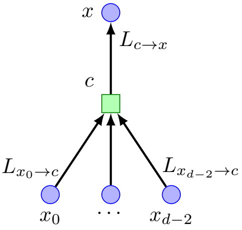
Plan

Propriétés

- **Théorème de concentration :** Les performances des codes longs d'un même ensemble (avec les mêmes $\lambda(X)$ et $\rho(X)$) se comportent globalement de la même manière $\Rightarrow$ on peut réaliser une étude en moyenne.
- **Pour des codes longs**, étude moyenne $\Leftrightarrow$ étude des codes acycliques ayant les mêmes $\lambda(X)$ et $\rho(X)$
- **Pour des codes longs**, effet de seuil sur le **paramètre du canal** (probabilité d'erreur, probabilité d'effacement ou SNR)

Évolution de densité - Nœuds de parités

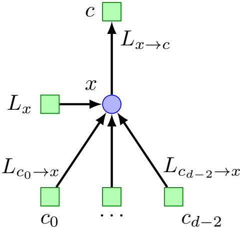
$$L_{c\to x} = 2\,\mathrm{atanh}\left(\prod_{k=0}^{d-2}\tanh\frac{L_{x_k\to c}}{2}\right)$$



- Canal BEC avec probabilité d'effacement p : $\mathbb{P}(y_i = \epsilon | x_i) = p$.
- **Remarque** Sur canal BEC les messages sont dans l'ensemble $\{\pm\infty, 0\}$.
- On note $\bar{p}_{x\to c}$ la probabilité d'effacement moyenne sur les messages allant des nœuds de variables aux nœuds de parités
- On suppose que $\mathbb{P}(L_{x_i\to c_j} = 0) = \bar{p}_{x\to c}, \forall i$ et j dans $\{0, n-1\}$ *et* $\{0, m-1\}$
- Que vaut $\bar{p}_{c\to x}$?

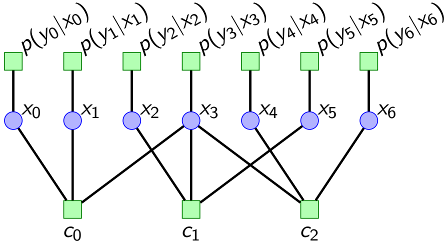
Évolution de densité - Nœuds de variables

$$L_{x\to c} = L_x + \sum_{k=0}^{d-2} L_{c_k\to x}$$



- Canal BEC avec probabilité d'effacement p : $\mathbb{P}(y_i = \epsilon | x_i) = p$.
- **Remarque** Sur canal BEC les messages sont dans l'ensemble $\{\pm\infty, 0\}$.
- On note $\bar{p}_{c\to x}$ la probabilité d'effacement moyenne sur les messages allant des nœuds de parités aux nœuds de variables
- On suppose que $\mathbb{P}(L_{c_j\to x_i} = 0) = \bar{p}_{c\to x}, \forall i$ et j dans $\{0, n-1\}$ *et* $\{0, m-1\}$
- Que vaut $\bar{p}_{x\to c}$?

Évolution de densité - Points fixes



- Canal BEC avec probabilité d'effacement p : $\mathbb{P}(y_i = \epsilon | x_i) = p$.
- La probabilité d'effacement après décodage pour $\ell \to \infty$ dépend des points fixes de

$$\bar{p}_{x\to c}^{(\ell)} = p\lambda\left(1 - \rho\left(1 - \bar{p}_{x\to c}^{(\ell-1)}\right)\right)$$