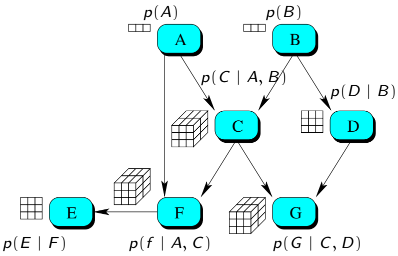
# Rappels : Réseaux bayésiens




### Factorisation dans un BN

$P(X_1, \cdots, X_n) = \prod_i P(X_i \mid \text{parents}_{X_i})$

Inférences dans un BN

- $P(A|G = 1)$
- $P(G|A = 0)$
- $P(E \mid A = 0, G = 1)$
- $\ldots$

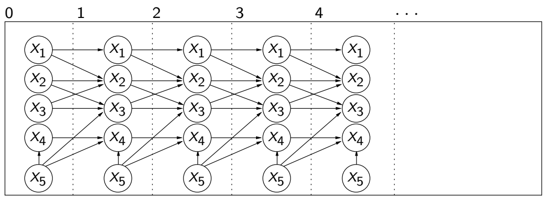


# Les réseaux bayésiens dynamiques

### dBN (dynamic BN)

Un réseau bayésien dynamique est un réseau bayésien dont les variables sont indicées par le temps $t$ et par $i$ : $X^{(t)} = X_1^{(t)}, \cdots, X_N^{(t)}$ et dont la distribution vérifie certaines propriétés :

- Markov ordre 1 :
  $P(X^{(t)} \mid X^{(0)}, \cdots, X^{(t-1)}) = P(X^{(t)} \mid X^{(t-1)})$,
- Homogénéité :
  $P(X^{(t)} \mid X^{(t-1)}) = \cdots = P(X^{(1)} \mid X^{(0)})$.




- L'adjectif *dynamique* n'est pas forcément bien choisi (puisqu'il y a homogénéité). *Temporel* ou *séquentiel* eût été de meilleur goût.
- On remarque que d'après la définition, les arcs d'un dBN vont de $X^{(t-1)}$ à $X^t$ ou restent dans le même $X^t$ (le même *timeslice*).
- Formellement, un réseau bayésien dynamique peut être considéré comme virtuellement infini.
- La définition ci-dessus est celle d'un dBN du premier ordre ($t$ ne dépend que de $t-1$). On pourrait, bien évidemment, définir des dBNs d'ordre supérieur.



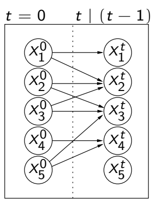
# Les réseaux bayésiens dynamiques (2-TBN)

### 2-TBN

Un réseau bayésien dynamique est défini :

- par les conditions initiales ($P(X^{(0)})$)
- par les relations entre des variables à l'instant $t-1$ et ces même variables à l'instant $t$ (*timeslice*).

Cette représentation, appelée 2TBN (2 timeslice BN) permet de modéliser un BN virtuellemet infini qui en est le développement dans le temps, à partir d'un instant 0.

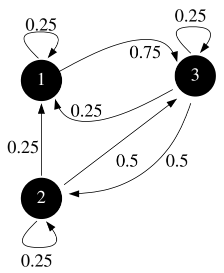



$P(x_1^{(t)}, \ldots, x_5^{(t)} \mid x_1^{(t-1)}, \ldots, x_5^{(t-1)})$

$= P(x_1^{(t)} \mid x^{(t-1)})$
$P(x_2^{(t)} \mid x_1^{(t-1)}, x_2^{(t-1)}, x_3^{(t-1)})$
$P(x_3^{(t)} \mid x_2^{(t-1)}, x_3^{(t-1)}, x_4^{(t-1)})$
$P(x_4^{(t)} \mid x_4^{(t-1)}, x_5^{(t-1)})$
$P(x_5^{(t)})$

1024 contre 4+16+16+8+2=46 ! !

# Chaîne de Markov et réseaux bayésiens dynamiques

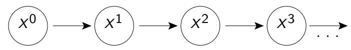


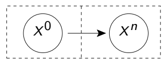

$$P(X^n \mid X^{n-1}) = \begin{pmatrix} 0.25 & 0 & 0.75 \\ 0.25 & 0.25 & 0.5 \\ 0.25 & 0.5 & 0.25 \end{pmatrix}$$

## Chaîne de Markov

- Une variable d'état discrète ($X^n$) (à l'instant $n$).
- Paramètres du modèle :
  - Condition initiale : $P(X^O)$
  - Modèle de transition : $P(X^n \mid X^{n-1})$

Réseau bayésien dynamique équivalent :

dBN : 


2TBN : 

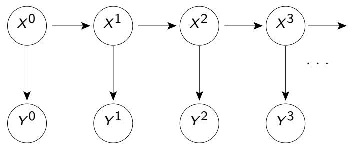


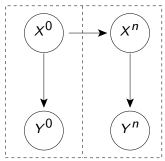

# HMM et réseaux bayésiens dynamiques

## HMM simple

- Une variable d'état discrète ($X^n$) (à l'instant $n$).
- Une variable d'observation discrète ($Y^n$)
- Paramètres du modèle :
  - Condition initiale : $P(X^O)$
  - Modèle de transition : $P(X^n \mid X^{n-1})$
  - Modèle d'observation : $P(Y^n \mid X^n)$

Ce qui donne, modélisé comme un réseau bayésien dynamique :

dBN : 


2TBN : 




# Inférences dans les réseaux bayésiens dynamiques

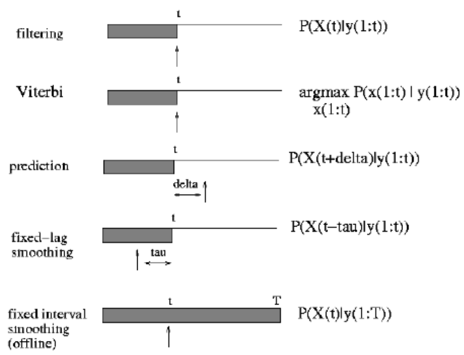
- A priori, très complexe : nombre de nœuds importants
- A priori, très complexe : “causes” communes dans un passé (lointain)

## Complexité

NP-difficile

## $P(X_t^i \mid y_{[1:r]})$ ?

- $r = t$ : *Filtering*
- $r > t$ : *Smoothing*
- $r < t$ : *Prediction*
- MPE : *Viterbi*



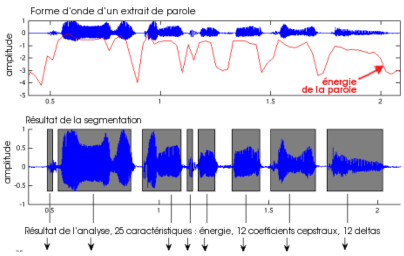
# Traitement des séquences

**Tâches :**

- Classification / Clustering
- Etiquetage / Segmentation
- Génération de séquences

- Reconnaissance de chaîne de caractères


- Reconnaissance de paroles






# Traitement des séquences : *structured data*

**Gene prediction**

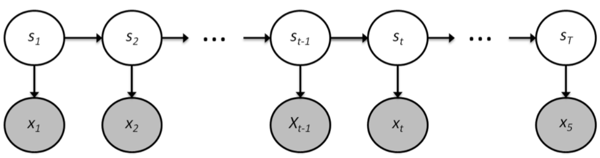
Find sequences of bases that encode proteins.

Soit $x = (x_1, \cdots, x_n)$ un séquence de bases ADN ($x_i \in \{a, c, t, g\}$).
Chaque séquence $x$ a une séquence associée de *labels*
$y = (y_1, \cdots, y_n), y_i \in \{C_1, C_2\}$ (codant ou non codant).

Aron Culotta, David Kulp and Andrew McCallum. Gene Prediction with Conditional Random Fields. Technical Report UM-CS-2005-028. University of Massachusetts, Amherst, 2005.



# Modèles de Markov Cachés (MMC = HMM)




- Séparation des **observations** et des **états**
  - Une chaîne = une séquence d'observations...
  - ... chaque observation ayant été générée par un état
- 2 composantes
  - Chaîne de Markov d'états finis
  - Ensemble de lois de probabilité d'émission
- 2 points de vues
  - Génératif : la chaîne génère des états qui entraînent des observations
  - Analytique : les observations fournissent de l'évidence sur la suite d'états (décodage) et sur le modèle (apprentissage)

# (Possibles) Notations

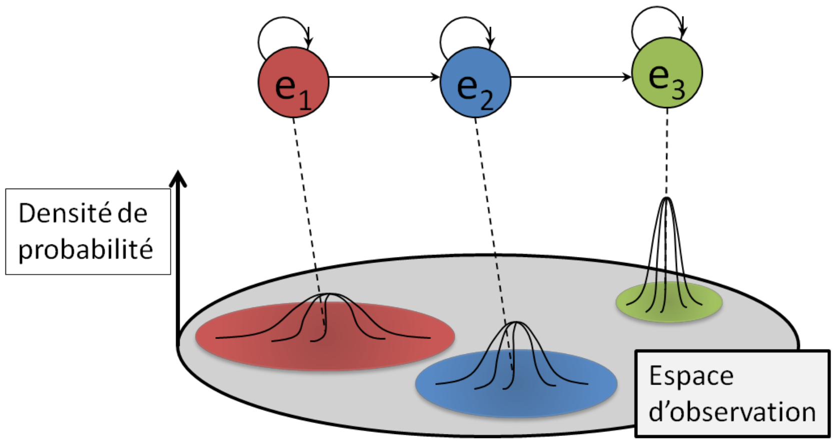
- La chaîne de Markov est toujours composée de :
  - d'une séquence d'**états** $S = (s_1, \dots, s_T)$
  - dont les valeurs sont tirées dans un ensemble fini $Q = (q_1, ..., q_N)$
  - Le modèle est toujours défini par $\{\Pi, A\}$
    - $\pi_i = P(s_1 = q_i)$
    - $a_{ij} = p(s_{t+1} = q_j | s_t = q_i)$
- Les **observations** sont modélisées à partir des $s_t$ :
  - séquence d'observation : $X = (x_1, \dots, x_T)$
  - loi de probabilité : $b_j(t) = p(x_t | s_t = q_j)$
  - $B$ peut être discrète ou continue
- MMC : $\lambda = \{\Pi, A, B\}$

**Ce qu'on manipule**

- Séquence d'observations : $X = (x_1, \dots, x_T)$
- Séquence d'états **(cachée = manquante)** : $S = (s_1, \dots, s_T)$



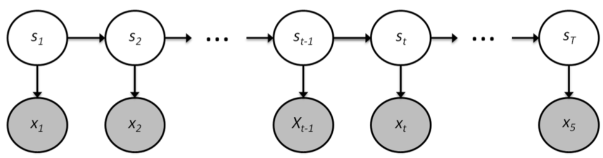
# MMC à lois d'observation continues




La figure présente un modèle *Gauche/Droite*... Quelle est sa particularité ?



# Hypothèses MMC




- CM d'ordre 1 :

$$p(s_t | s_1, \dots, s_{t-1}) = p(s_t | s_{t-1})$$

- Indépendance des observations :

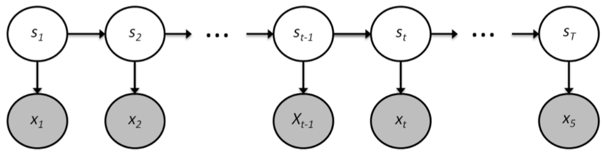
$$p(x_t | x_{t-1}, s_t) = p(x_t | s_t)$$

i.e. : les observations successives sont non-corrélées !

**Notations simplifiées (de $a$ à $b$) :**

$$s_1^t = s_1, \dots, s_t, \quad x_1^t = x_1, \dots, x_t$$

# Validation...




Comment calculer probabilités suivantes ?

- Séquence d'états
$$p(s_1^T|\lambda) = p(s_1, \dots, s_T|\lambda)$$
- Séquence d'observations (connaissant les états)
$$p(x_1^T|s_1^T, \lambda)$$
- Séquences obs + états :
$$p(x_1^T, s_1^T|\lambda) = p(x_1, \dots, x_T, s_1, \dots, s_T|\lambda)$$



# Point de vue génératif

**Génération d'une séquence $\{x_1, \dots, x_T\}$**

**Data** : $A, B, \Pi$
**Result** : $x_1^T, s_1^T$

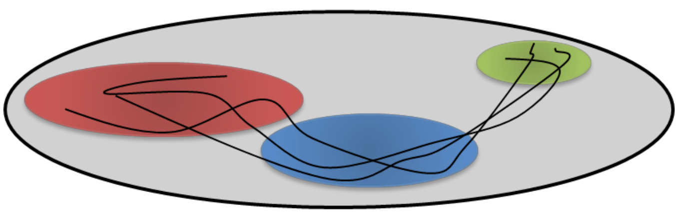
$S \leftarrow []$;
$X \leftarrow []$;
Tirer $s_1$ en fonction de $\Pi$;
Tirer $x_1$ en fonction de $B$ et $s_1$;
$s_t \leftarrow s_1$, $x_t \leftarrow x_1$, $t \leftarrow 1$;
$S \leftarrow [S,\ s_t]$, $X \leftarrow [X,\ x_t]$;
**while** *$s_t$ n'est pas l'état final* **do**
  $s_{t+1}$ ←tirage selon $(A(s_t, :))$;
  $x_{t+1}$ ←tirage selon $(B(s_{t+1}, :))$;
  $s_t \leftarrow s_1$, $x_t \leftarrow x_1$;
  $S \leftarrow [S,\ s_t]$, $X \leftarrow [X,\ x_t]$;
  $t \leftarrow t + 1$;



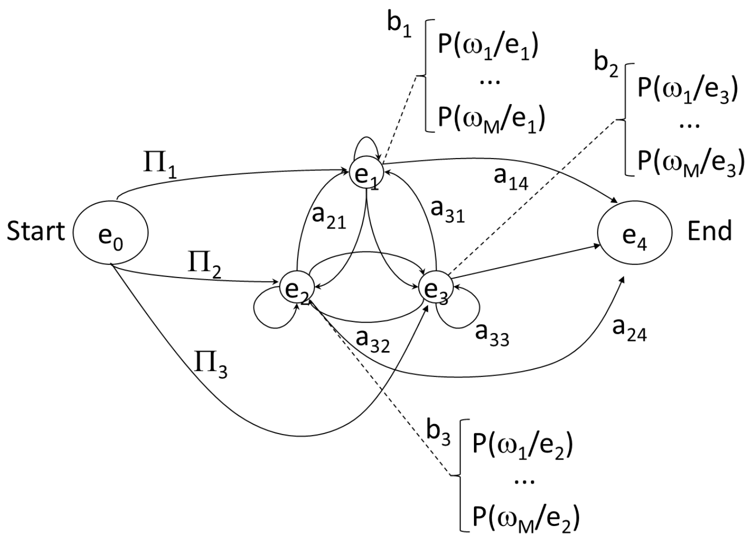
# Représentation d'une trajectoire

- trajectoire d'états


- trajectoire d'observations

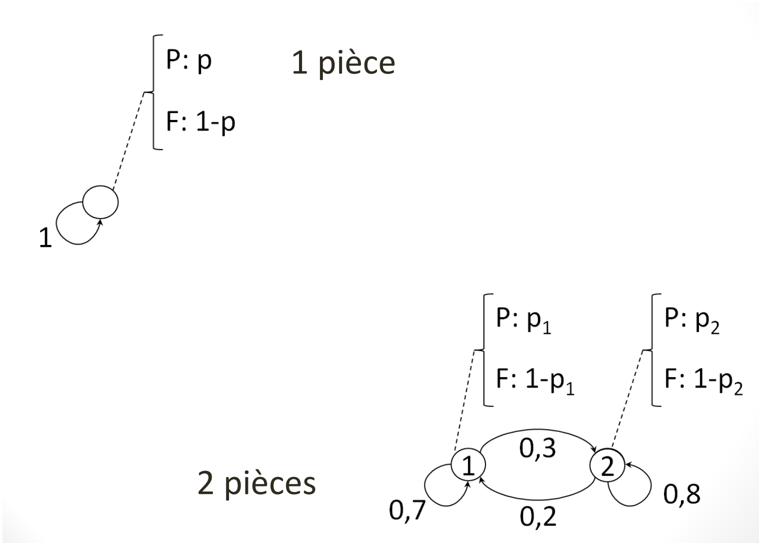


# Représentation d'un MMC





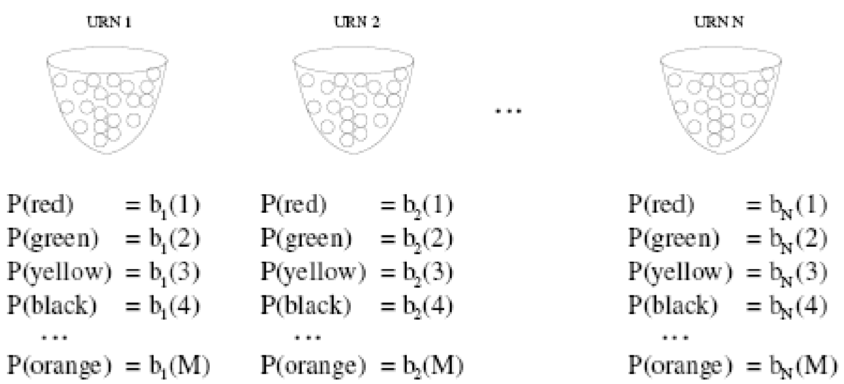

# Modélisation des pièces






# Capacités d'expression CM, MMC

- *N* urnes (de Rabiner)
  - chaque urne = distribution spécifique des couleurs de boules
- On tire successivement des boules dans les différentes urnes et on note les couleurs des boules



## Usage des MMC

- Modélisation complexe
  - mixture de sources émettrices
- Problèmes de séquençage des signaux



## Les trois problèmes des MMC (Fergusson - Rabiner)

- **Evaluation** : $\lambda$ donné, calcul de $p(x_1^T|\lambda)$
- **Décodage** : $\lambda$ donné, quelle séquence d'états a généré les observations ?

$$s_1^{T\star} = \arg\max_{s_1^T} p(x_1^T|\lambda)$$

- **Apprentissage** : à partir d'une série d'observations, trouver $\lambda^\star$

$$\lambda^\star = \{\Pi^\star, A^\star, B^\star\} = \arg\max_{s_1^T, \lambda} p(x_1^T|\lambda)$$



## PB1 : évaluation $p(x_1^T|\lambda)$

- Quel point de départ ? Proba totales

$$p(x_1^T|\lambda) = \sum_{s_1^T} p(x_1^T, s_1^T|\lambda)$$

- Simplifier la combinatoire... Hypothèse MMC ordre 1

$$p(x_1^T|\lambda) = \sum_{s_1^T} \prod_{t=1}^{T} p(s_t|s_{t-1}) p(x_t|s_t)$$

- Complexité ? $\mathcal{O}(N^T)$

À chaque pas de temps, envisager toutes les combinaisons d'états qui ont pu amener ici...

# PB1 : évaluation $p(x_1^T|\lambda)$

**Calcul par récurrence :**

- On pose :

$$\alpha_t(i) = p(x_1^t, s_t = i|\lambda)$$

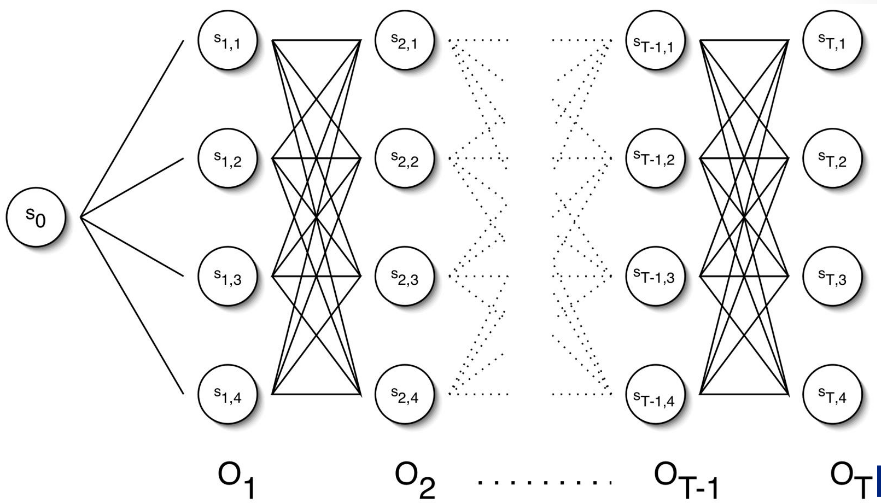
- Que vaut alors $p(x_1^T|\lambda)$ ?

$$p(x_1^T|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$



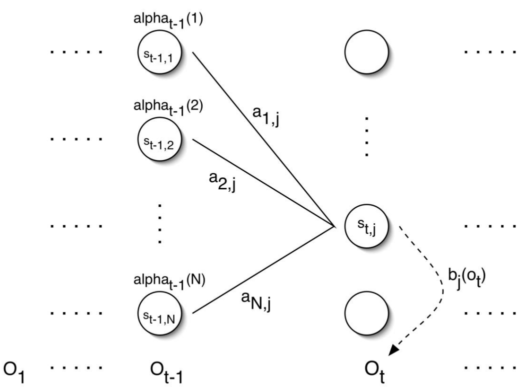
# PB1 : Treillis

**Envisager tous les états à tous les pas de temps (+ transitions) = trellis d'hypothèses**






# PB1 : Algorithme *forward*




$$\alpha_t(j) = p(x_1^t, s_t = j|\lambda)$$

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j(x_t)$$

Formalisation récursive = briser la complexité

# PB1 : Algorithme *forward*

- Initialisation :
$$\alpha_{t=1}(i) = p(x_1^1, s_1 = i|\lambda) = \pi_i b_i(x_1)$$
- Itération :
$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j(x_t)$$
- Terminaison :
$$p(x_1^T|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$
- Complexité linéaire en $T$
  - Usuellement : $T >> N$



# PB2 : décodage

$$s_1^{T\star} = \arg\max_{s_1^T} p(x_1^T|\lambda)$$

Avec la formule précédente (hyp. MMC ordre 1) :

$$s_1^{T\star} = \arg\max_{s_1^T} \prod_{t=1}^{T} p(s_t|s_{t-1})p(x_t|s_t)$$

- Par défaut, très complexe
- Même schéma que précédemment = **algorithme de Viterbi**



# PB2 : Viterbi

- On pose :
  Meilleur score pour un chemin au temps $t$, se terminant à l'état $i$ :
$$\delta_t(i) = \max_{s_1^{t-1}} p(s_1^{t-1}, s_t = i, x_1^t|\lambda)$$

A $t$, la probabilité du meilleur chemin est la combinaison :
- d'un des meilleurs chemins précédents...
- ... et de la transition vers $s_j$ + observation de $x_t$ à partir de $s_j$

- Initialisation :
$$\delta_1(i) = \pi_i b_i(x_1)$$
- Récurrence :
$$\delta_t(j) = \left[\max_i \delta_{t-1}(i) a_{ij}\right] b_j(x_t)$$

## PB2 : Viterbi (suite)

- $\delta$ = stockage de la probabilité de certaines situations...
- Ce qui nous intéresse c'est la séquence d'états associée $\Rightarrow$ stockage à prévoir

- Stockage des indices des états dans un tableau $\Psi$

$$\Psi_t(j) = \arg \max_{i \in [1,\ N]} \delta_{t-1}(i) a_{ij}$$

Pour arriver en $j$ à $t$, quel était le meilleur état à $t-1$ ???

- $\Rightarrow$ A parcourir à l'envers pour retrouver le chemin optimal !
- Complexité en $\mathcal{O}(N^2 T)$



## PB2 : Viterbi (récapitulatif)

$$\delta_t(i) = \max_{s_1^{t-1}} p(s_1^{t-1}, s_t = i, x_1^t | \lambda)$$

1. Initialisation
$$\begin{aligned} \delta_1(i) &= \pi_i b_i(x_1) \\ \Psi_1(i) &= 0 \end{aligned}$$
2. Récursion
$$\begin{aligned} \delta_t(j) &= \left[\max_i \delta_{t-1}(i) a_{ij}\right] b_j(x_t) \\ \Psi_t(j) &= \arg \max_{i \in [1,\ N]} \delta_{t-1}(i) a_{ij} \end{aligned}$$
3. Terminaison
$$S^\star = max_i \delta_T(i)$$
4. Chemin
$$\begin{aligned} q_T^\star &= \arg \max_i \delta_T(i) \\ q_t^\star &= \Psi_{t+1}(q_{t+1}^\star) \end{aligned}$$