Complexité CM9

Antonio E. Porreca
aeporreca.org/complexite

Diapositives basées sur le livre Complexité algorithmique par Sylvain Perifel

Précédemment dans Complexité…

⚠️ Définition 3-A (p. 64) ⚠️

Réductions (many-one) polynomiales

- Une réduction (many-one) en temps polynomial d'un problème L_1 (sur l'alphabet Σ_1) à un problème L_2 (sur l'alphabet Σ_2) est une fonction $f\colon\ \Sigma_1^\star \to \Sigma_2^\star$ calculable en temps polynomial telle que

$$\forall x \in \Sigma_1^\star \quad x \in L_1 \iff f(x) \in L_2$$

- Si une telle f existe, on dit que L_1 se réduit à L_2 (via f) et on notera $L_1 \leq_{\mathrm{m}}^{\mathbf{P}} L_2$ (ou parfois, en bref, $L_1 \leq L_2$)

⚠️ Définition 3-J (p. 67) ⚠️

Difficulté et complétude

Soit L un problème et $\mathscr{C}$ une classe de complexité

- On dit que L est $\mathscr{C}$-difficile (ou $\mathscr{C}$-dur) si pour tout problème $L' \in \mathscr{C}$ on a $L' \leq L$

- On dit que L est $\mathscr{C}$-complet s'il est $\mathscr{C}$-difficile et en plus on a $L \in \mathscr{C}$

Proposition 3-M (p. 68)

La prédiction est $\mathbf{NP}$-complète

Le problème suivant est $\mathbf{NP}$-complet :

$$A = \{(\langle N\rangle, x, 1^t) : N(x) \text{ accepte en temps } \leq t\}$$

code d'une MT non deterministe

mot d'entrée pour la MT N

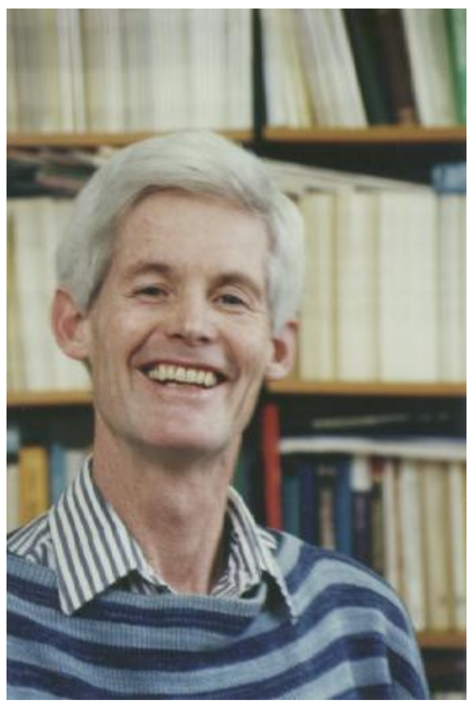
entier $t \in \mathbb{N}$ en unaire

Et maintenant,
la suite

Le théorème de Cook-Levin, ou : Enfin, un problème $\mathbf{NP}$-complet intéressant ! 🎉

⚠️ Théorème 3-V (p. 72) ⚠️

Cook 1971, Levin 1973



Stephen Cook

Леони́д Ле́вин

SAT est $\mathbf{NP}$-complet

⚠️ Théorème 3-V (p. 72) ⚠️

Cook 1971, Levin 1973

Le fonctionnement en temps polynomial d'une machine non déterministe N sur une entrée x est décrit par une formule φ calculable en temps polynomial telle que le nombre d'affectations satisfaisant φ est égal au nombre de chemins acceptants de $N(x)$.

Idée de la démonstration

- SAT $\in$ **NP** car il suffit de deviner une assignation des variables et vérifier en temps polynomial qu’elle satisfait la formule

- La complétude vient du fait qu’on peut décrire par une formule de taille polynomiale le diagramme espace-temps d’une exécution d’une machine non déterministe polynomiale car celui-ci répond à des règles locales

Idée de la démonstration

- En d'autres termes, on décrit par une formule $\varphi(y)$ le fonctionnement de la machine le long du chemin (découlant du choix des transitions) décrit par y

- Pour savoir s'il existe un chemin acceptant dans le calcul de la machine, il suffit alors de savoir s'il existe une affectation des variables y de la formule pour laquelle l'état final du diagramme décrit est acceptant, ce qui est un problème de type SAT

Démonstration : SAT $\in \mathbf{NP}$ 😴

- Algo non déterministe pour SAT sur l'entrée $\varphi(x_1, \ldots, x_n)$:
    - deviner $(a_1, \ldots, a_n) \in \{0,1\}^n$
    - accepter ssi $\varphi(a_1, \ldots, a_n) = 1$
- En alternative, un vérificateur déterministe sur l'entrée $\left(\varphi(x_1, \ldots, x_n), a_1, \ldots, a_n\right)$:
    - accepter ssi $\varphi(a_1, \ldots, a_n) = 1$

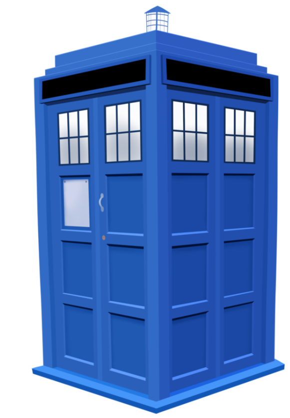
Démonstration : $B \leq$ SAT pour tout $B \in \mathbf{NP}$ 😎

- Soit $B \in \mathbf{NP}$
- À toute instance x de B on associe une formule φ_x …
- …telle que φ_x est satisfaisable ssi $x \in B$
- Les variables de φ_x désigneront en quelque sorte le chemin de calcul à suivre

Démonstration :
$B \leq$ SAT pour tout $B \in \mathbf{NP}$ 😎

- Soit N une machine non déterministe qui reconnaît B
  - en temps polynomiale $p(n)$
  - avec ensemble d'états Q, alphabet de travail Γ
- nous allons « simuler » le fonctionnement de N le long d'un chemin arbitraire par φ_x
- Pour cela, nous allons considérer le diagramme espace-temps de $N(x)$

Diagramme espace-temps de la MT N sur l'entrée $x = x_1 \cdots x_n$



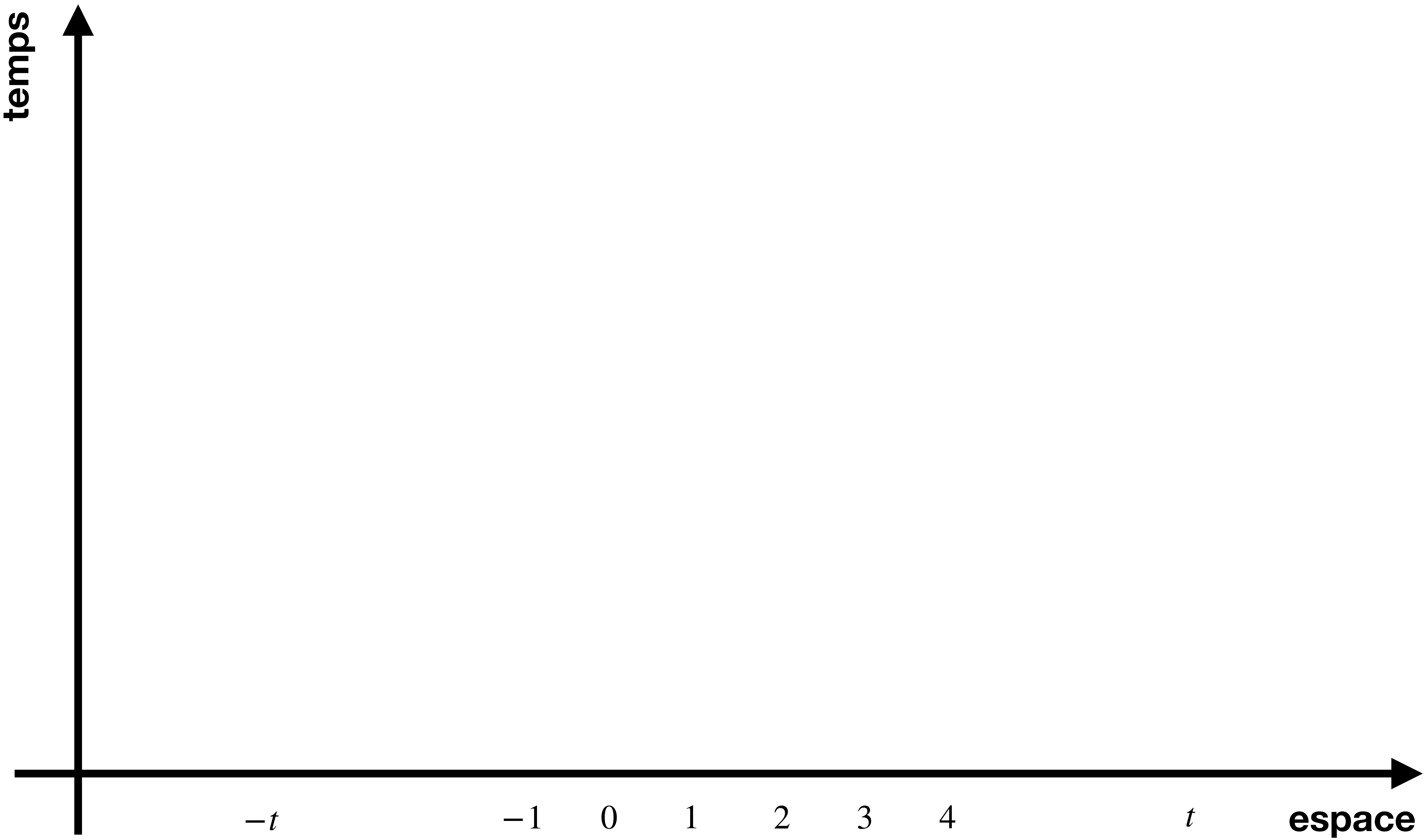
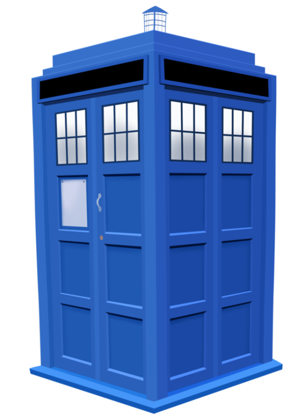


Diagramme espace-temps de la MT N sur l'entrée $x = x_1 \cdots x_n$



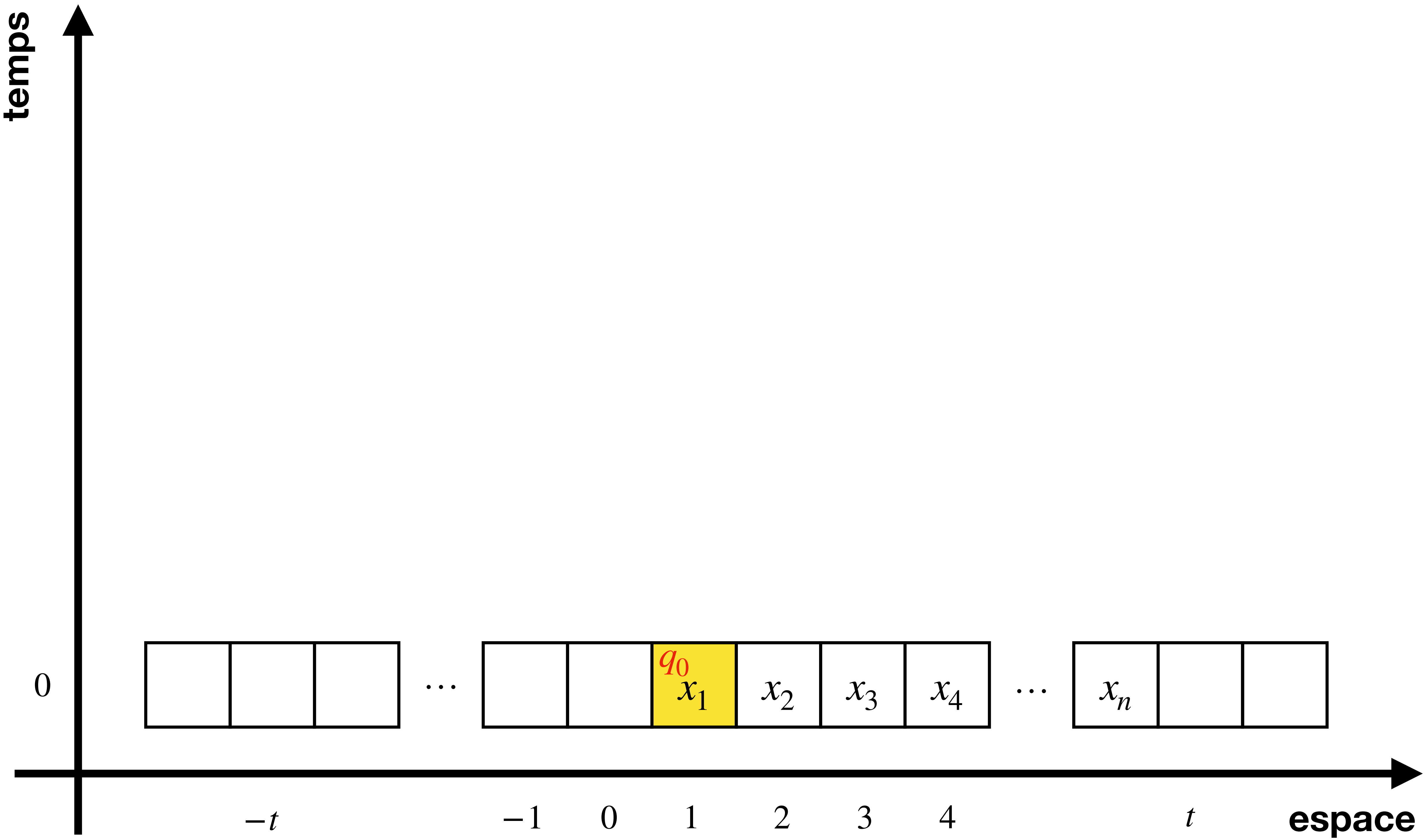
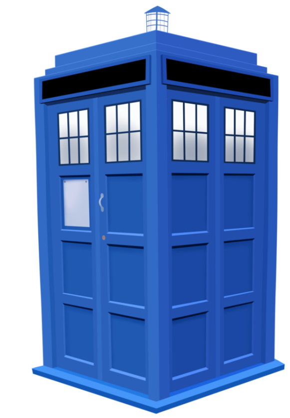


Diagramme espace-temps de la MT N sur l'entrée $x = x_1 \cdots x_n$



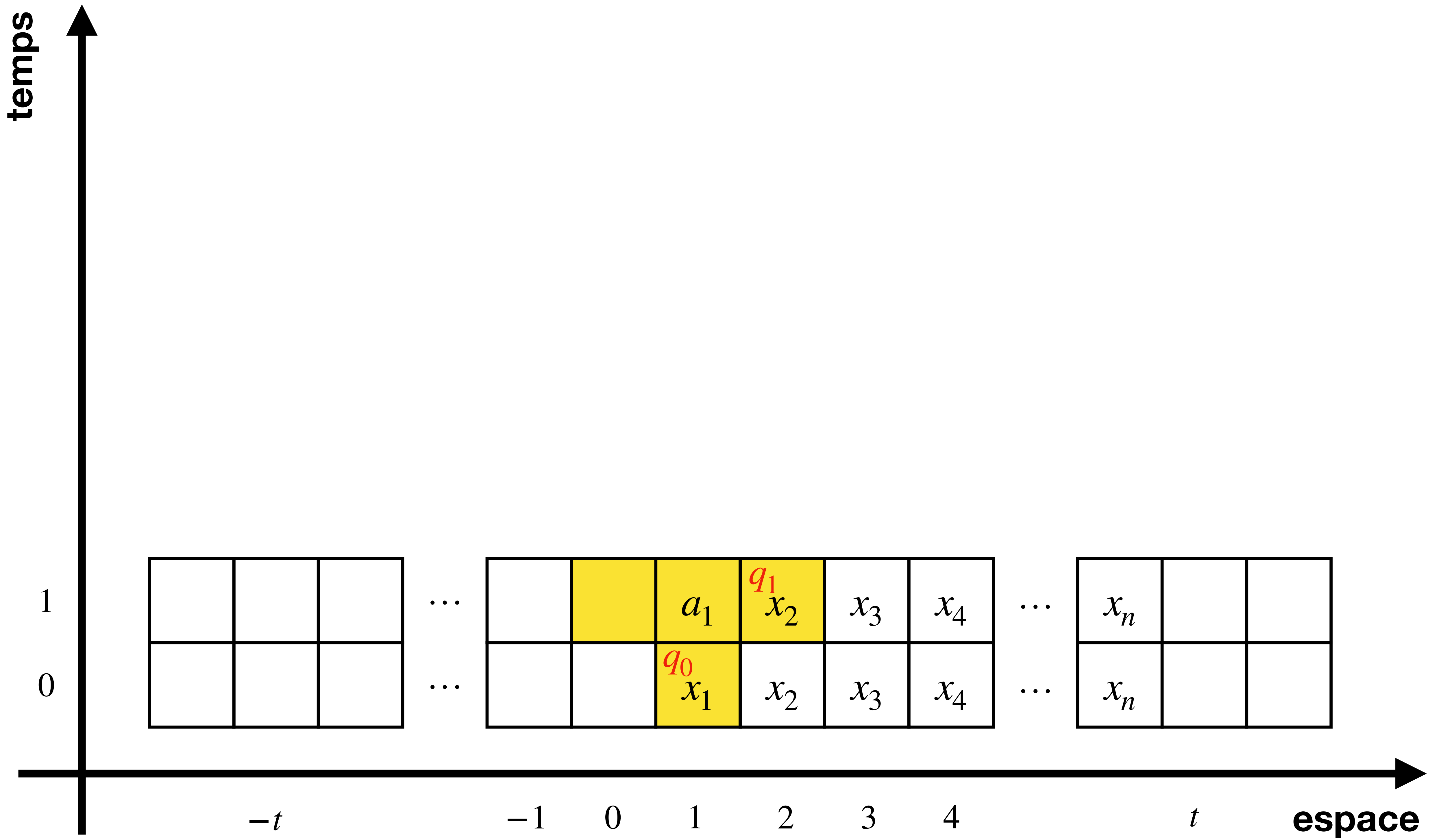
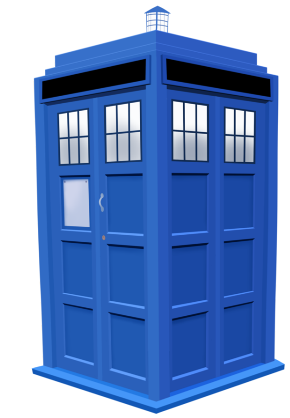


Diagramme espace-temps de la MT N sur l'entrée $x = x_1 \cdots x_n$



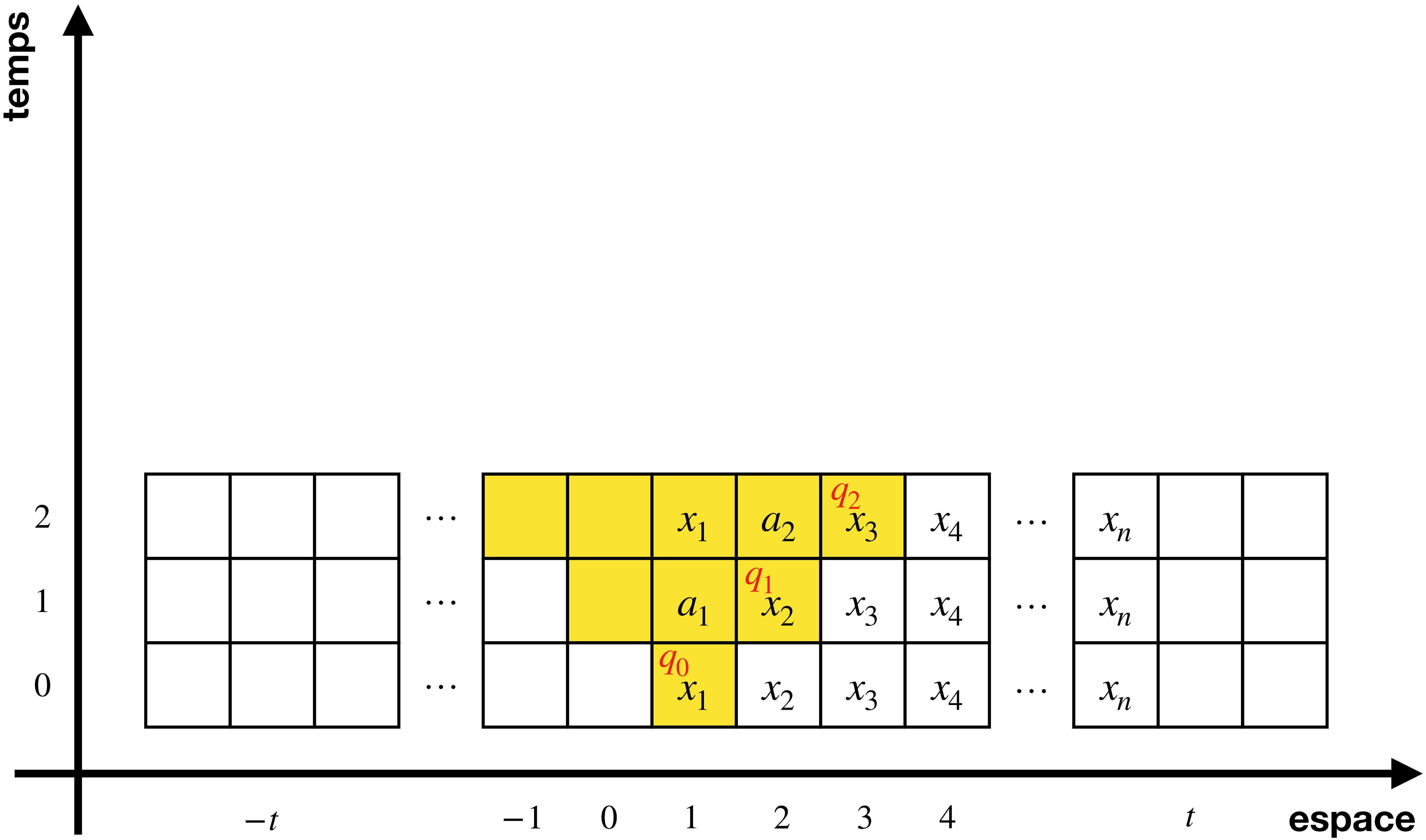
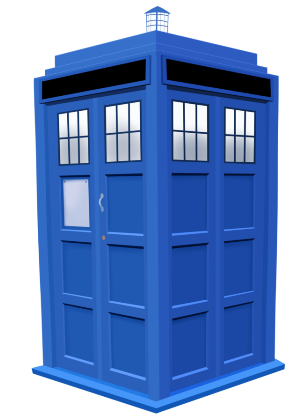


Diagramme espace-temps de la MT N sur l'entrée $x = x_1 \cdots x_n$



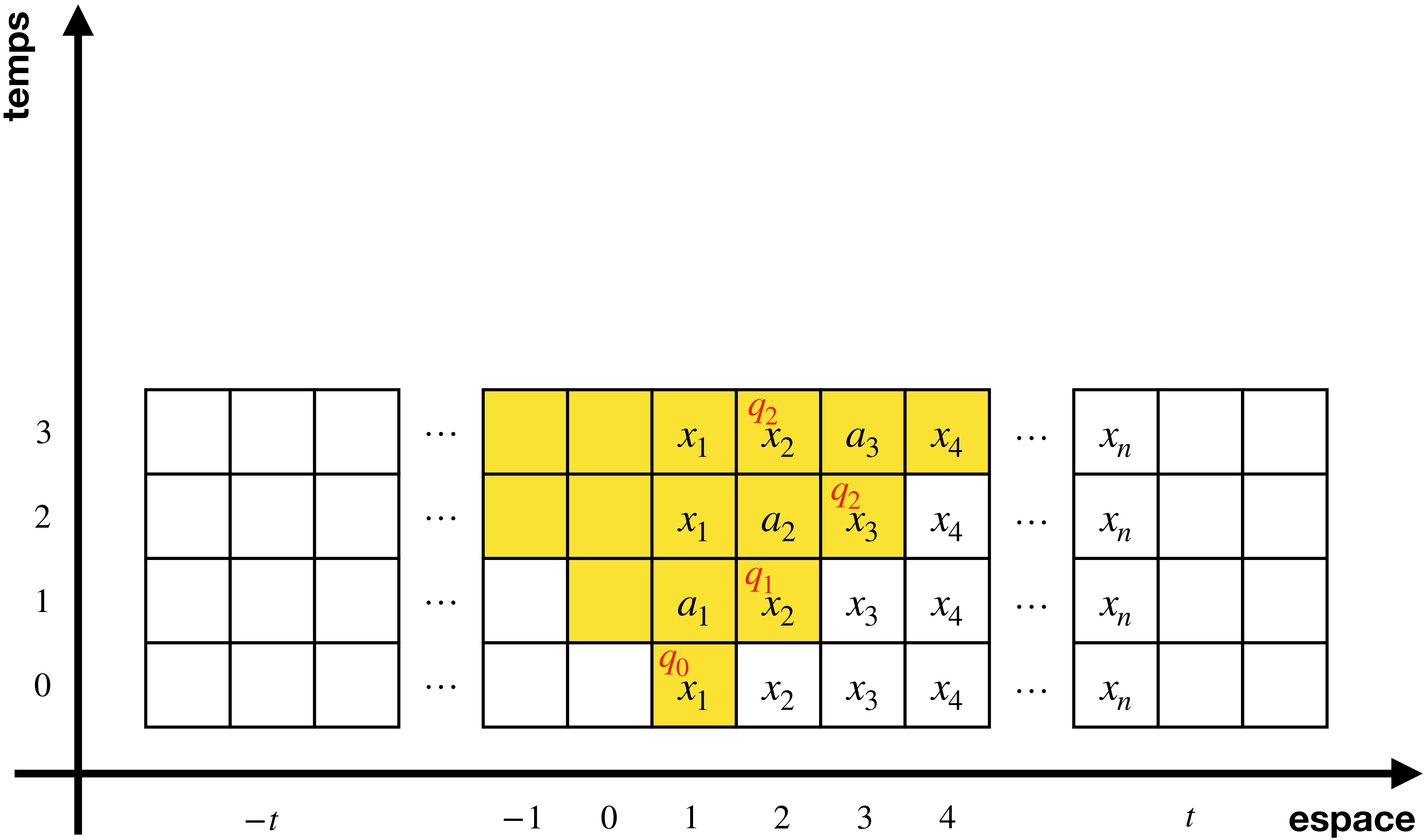
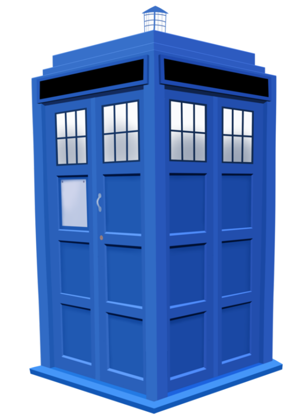


Diagramme espace-temps de la MT N sur l'entrée $x = x_1 \cdots x_n$



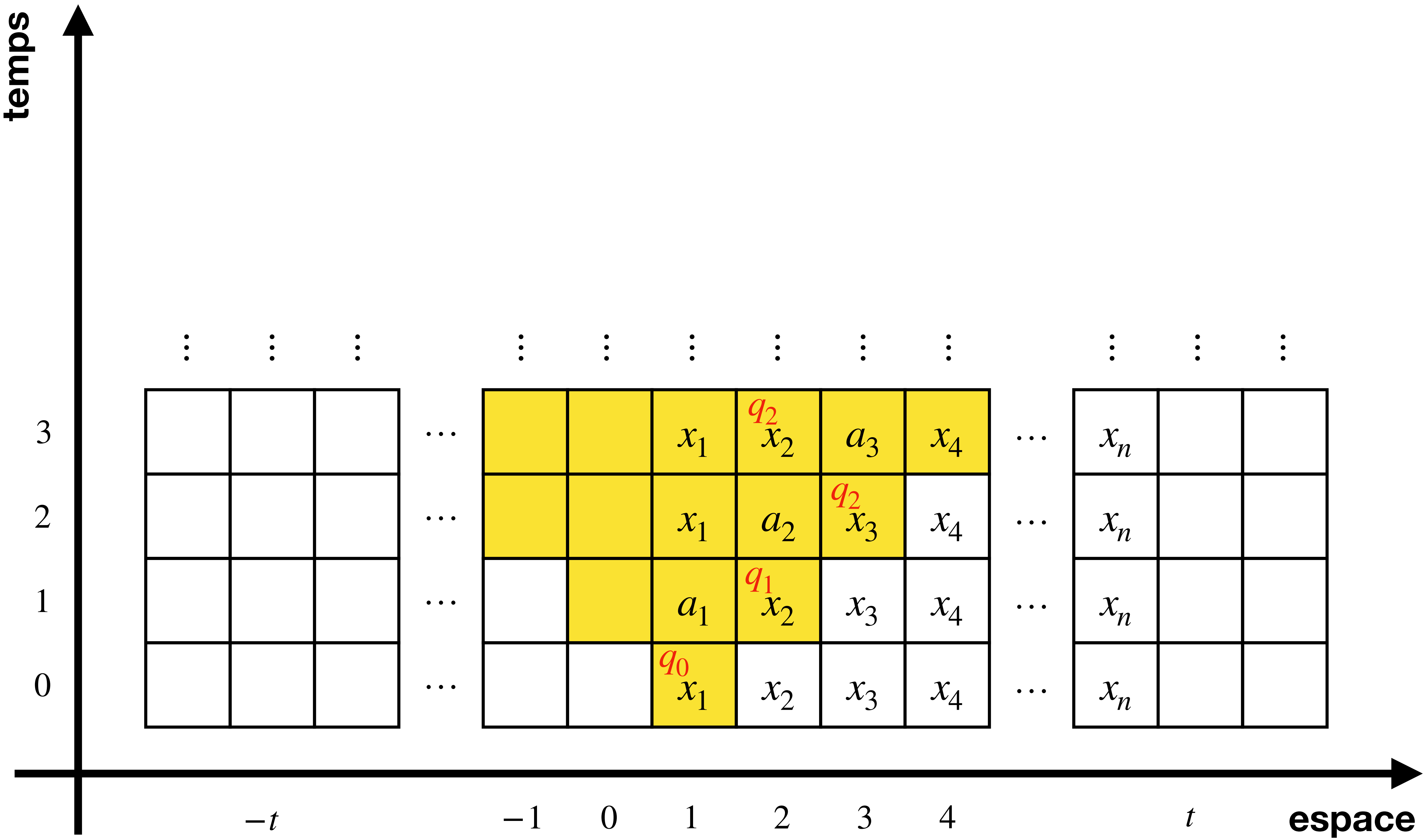
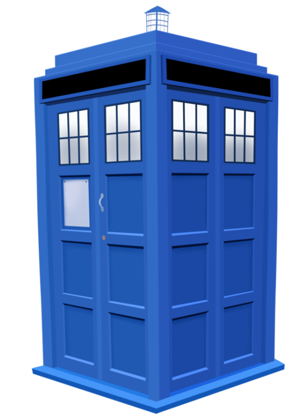


Diagramme espace-temps de la MT N sur l'entrée $x = x_1 \cdots x_n$



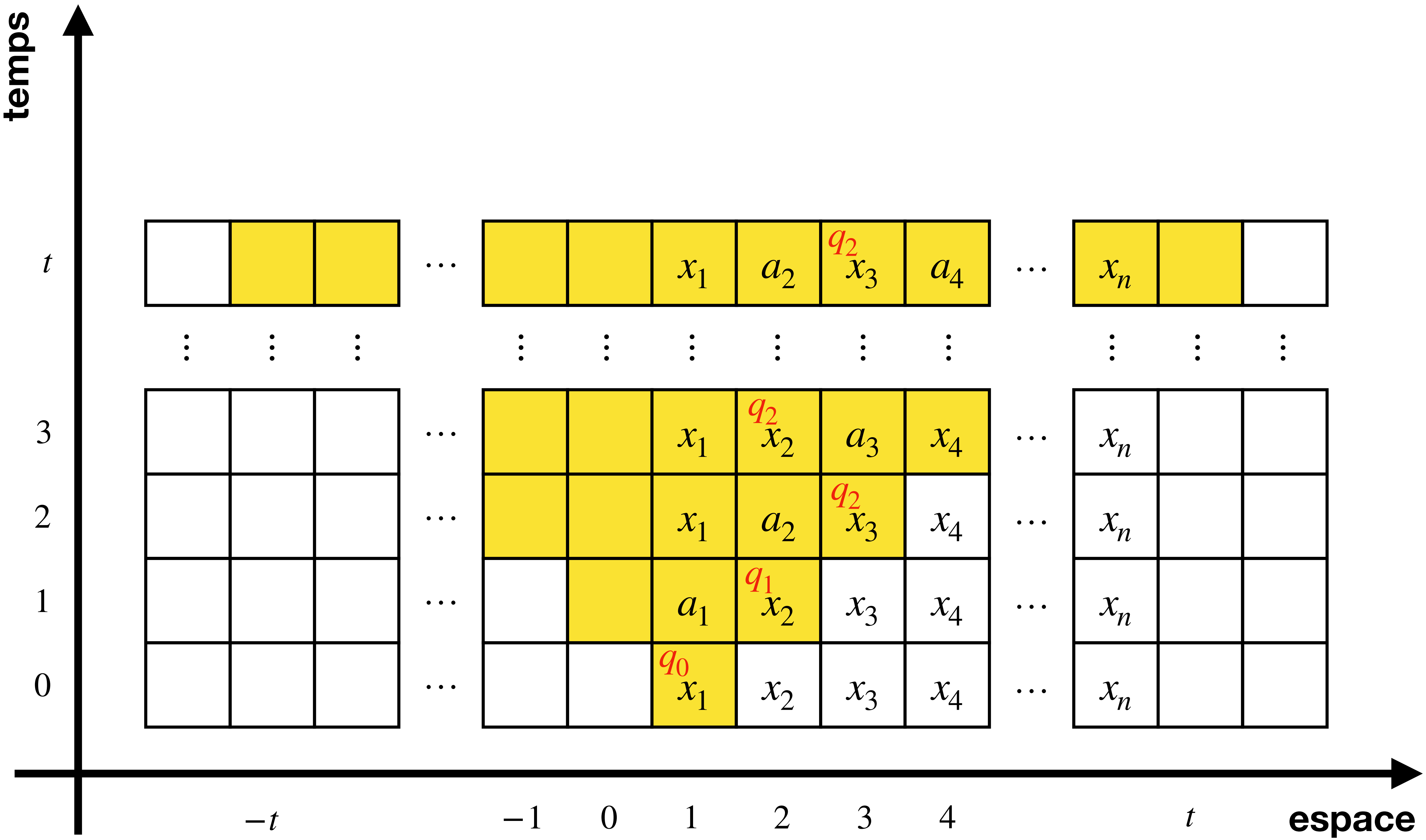
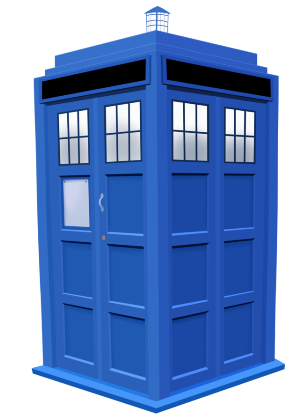


Diagramme espace-temps de la MT N sur l'entrée $x = x_1 \cdots x_n$



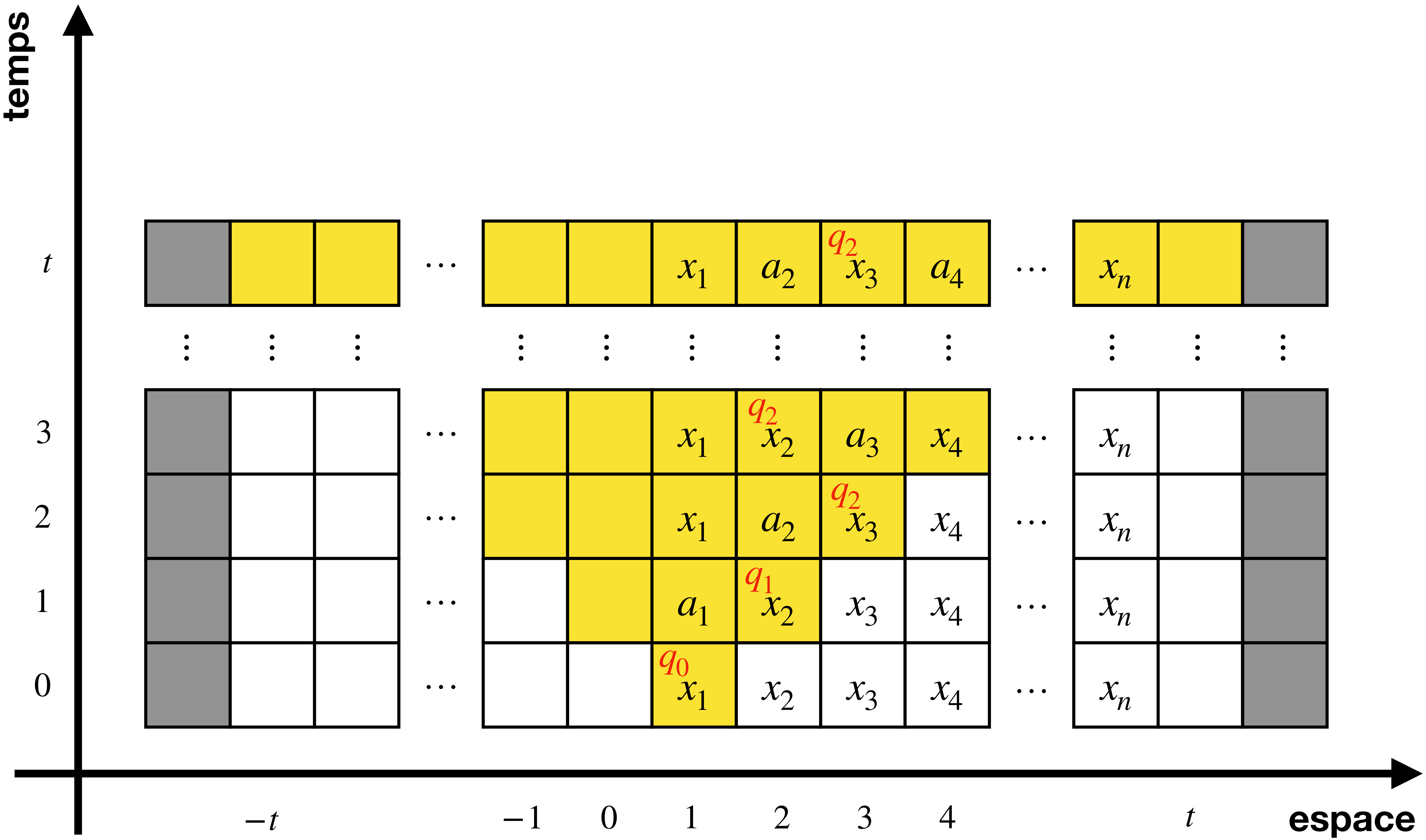


Diagramme espace-temps de la MT N sur l'entrée $x = x_1 \cdots x_n$

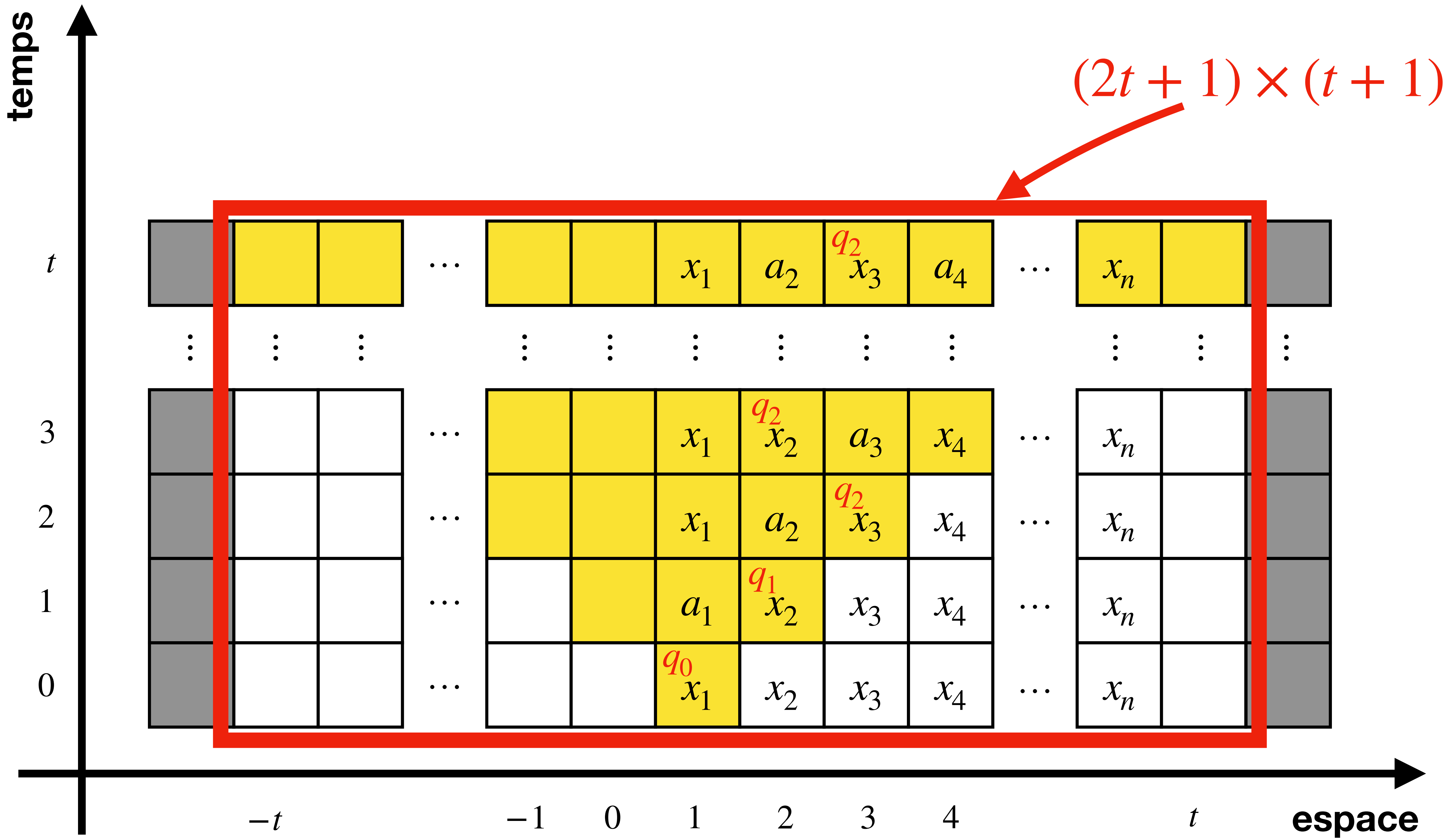
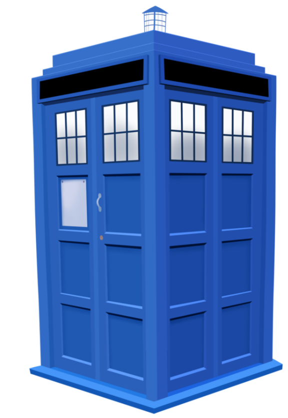


Diagramme espace-temps de la MT N sur l'entrée $x = x_1 \cdots x_n$



- Le diagramme espace-temps du ruban de N décrit le calcul sur l'entrée x
- Si le calcule de $N(x)$ termine en moins de t étapes, on suppose que le diagramme espace-temps répète la dernière ligne
- Notre formule φ_x va affirmer que le calcul commence dans la configuration initiale, que le diagramme espace-temps de la machine est cohérent avec la relation de transition et qu'on termine dans un étant acceptant

La formule φ_x a quatre parties

- cohérence, signifiant que deux valeurs ne sont pas assignés à la même case, deux positions à la même tête, deux états à la même étape
- début$_x$, signifiant que la configuration initiale est la bonne
- pour chaque étape j : transition$_j$, signifiant que la j-ème transition est valide
- accepte, signifiant qu'on arrive dans un état acceptant à un temps $\leq t$, ou $t = p(n)$

Variables de la formule φ_x

Pour chaque étape $j \in \{0,\ldots,t\}$, symbole $\gamma \in \Gamma$,
position $i \in \{-t,\ldots,0,\ldots,t\}$, état $q \in Q$:

- $c_{\gamma,i,j} = 1$ ssi la i-ème case du ruban contient le symbole γ au temps j
- $p_{i,j} = 1$ ssi la tête du ruban est à la position i au temps j
- $e_{q,j} = 1$ ssi l'état de la machine est q à l'instant j

En total on a $(t+1)(2t+1)|\Gamma| + (t+1)(2t+1) + (t+1)|Q|$ variables, ce qui est polynomial par rapport à x

⚠️ Notation ⚠️

Dans les formules logiques suivantes, les symboles

$$\bigwedge_{i=1}^{n} \phi_i \qquad \bigvee_{i=1}^{n} \phi_i$$

représentent succinctement les conjonctions ou disjonctions

$$\phi_1 \wedge \phi_2 \wedge \cdots \wedge \phi_n \qquad \phi_1 \vee \phi_2 \vee \cdots \vee \phi_n$$

c'est-à-dire, en réalité il faut répliquer explicitement les sous-formules φ_i pour toutes les valeurs de l'indice i

Par exemple :

$$\bigwedge_{i=1}^{2} \bigvee_{j=1}^{2} (x_i \wedge \neg y_j) = \big((x_1 \wedge \neg y_1) \vee (x_1 \wedge \neg y_2)\big) \wedge \big((x_2 \wedge \neg y_1) \vee (x_2 \wedge \neg y_2)\big)$$

Sous-formule cohérence

$$\bigwedge_{i,j} \bigvee_{\gamma} \left(c_{\gamma,i,j} \wedge \bigwedge_{\gamma' \neq \gamma} \neg c_{\gamma',i,j} \right)$$

$$\wedge$$

$$\bigwedge_{j} \bigvee_{i} \left(p_{i,j} \wedge \bigwedge_{i' \neq i} \neg p_{i',j} \right)$$

$$\wedge$$

$$\bigwedge_{j} \bigvee_{q} \left(e_{q,j} \wedge \bigwedge_{q' \neq q} \neg e_{q',j} \right)$$

Sous-formule cohérence

$$\bigwedge_{i,j} \bigvee_{\gamma} \left(c_{\gamma,i,j} \wedge \bigwedge_{\gamma' \neq \gamma} \neg c_{\gamma',i,j} \right)$$

à tout moment chaque case contient exactement un symbole

$$\wedge$$

$$\bigwedge_{j} \bigvee_{i} \left(p_{i,j} \wedge \bigwedge_{i' \neq i} \neg p_{i',j} \right)$$

$$\wedge$$

$$\bigwedge_{j} \bigvee_{q} \left(e_{q,j} \wedge \bigwedge_{q' \neq q} \neg e_{q',j} \right)$$

rappel : i est une position, j une étape, γ un symbole

Sous-formule cohérence

$$\bigwedge_{i,j} \bigvee_{\gamma} \left(c_{\gamma,i,j} \wedge \bigwedge_{\gamma' \neq \gamma} \neg c_{\gamma',i,j} \right)$$

à tout moment chaque case contient exactement un symbole

$$\wedge$$

$$\bigwedge_{j} \bigvee_{i} \left(p_{i,j} \wedge \bigwedge_{i' \neq i} \neg p_{i',j} \right)$$

à tout moment chaque tête a exactement une position

$$\wedge$$

$$\bigwedge_{j} \bigvee_{q} \left(e_{q,j} \wedge \bigwedge_{q' \neq q} \neg e_{q',j} \right)$$

rappel : i est une position, j une étape, γ un symbole

Sous-formule cohérence

$$\bigwedge_{i,j} \bigvee_{\gamma} \left(c_{\gamma,i,j} \wedge \bigwedge_{\gamma' \neq \gamma} \neg c_{\gamma',i,j} \right)$$

à tout moment chaque case contient exactement un symbole

$$\wedge$$

$$\bigwedge_{j} \bigvee_{i} \left(p_{i,j} \wedge \bigwedge_{i' \neq i} \neg p_{i',j} \right)$$

à tout moment chaque tête a exactement une position

$$\wedge$$

$$\bigwedge_{j} \bigvee_{q} \left(e_{q,j} \wedge \bigwedge_{q' \neq q} \neg e_{q',j} \right)$$

à tout moment on est dans exactement un état

rappel : i est une position, j une étape, γ un symbole

Sous-formule début_x

$$e_{q_0,0}$$

$$\land$$

$$\left(\bigwedge_{i\leq 0 \lor i>n} c_{B,i,0}\right) \land \left(\bigwedge_{1\leq i\leq n} c_{x_i,i,0}\right)$$

$$\land$$

$$p_{1,0}$$

Sous-formule début_x

$$e_{q_0,0}$$

l'état de départ est l'état initial q_0

$$\wedge$$

$$\left(\bigwedge_{i \leq 0 \vee i > n} c_{B,i,0} \right) \wedge \left(\bigwedge_{1 \leq i \leq n} c_{x_i,i,0} \right)$$

$$\wedge$$

$$p_{1,0}$$

rappel : i est une position, j une étape, γ un symbole

Sous-formule début_x

$$e_{q_0,0}$$

l'état de départ est l'état initial q_0

$$\wedge$$

$$\Big(\bigwedge_{i\leq 0 \vee i>n} c_{B,i,0}\Big) \wedge \Big(\bigwedge_{1\leq i\leq n} c_{x_i,i,0}\Big)$$

le ruban contient x entre les positions 1 et n et B ailleurs

$$\wedge$$

$$p_{1,0}$$

rappel : i est une position, j une étape, γ un symbole

Sous-formule début_x

$$e_{q_0,0}$$

l'état de départ
est l'état initial q_0

$$\wedge$$

$$\Big(\bigwedge_{i\le 0 \vee i>n} c_{B,i,0}\Big) \wedge \Big(\bigwedge_{1\le i\le n} c_{x_i,i,0}\Big)$$

le ruban contient x entre
les positions 1 et n et B ailleurs

$$\wedge$$

$$p_{1,0}$$

la tête de lecture
est en position 1 au temps 0

rappel : i est une position, j une étape, γ un symbole

Sous-formule accepte

- Pour alléger les notations qui suivent et par abus de notation on pose

$$\delta(q_{oui}, \gamma) = \{(q_{oui}, \gamma, 0)\}$$
$$\delta(q_{no}, \gamma) = \{(q_{no}, \gamma, 0)\}$$

- C'est-à-dire, quand on accepte ou rejette, les configurations suivantes restent identiques

- Du coup on a tout simplement accepte $= e_{q_{oui},t}$ (au temps t on est dans l'état acceptant q_{oui})

On est où ? 🧭

cohérence $\wedge$ **début**$_x$ $\wedge$ **accepte**

On est où ?

**deux symboles ne sont pas assignés
à la même case, deux positions
à la même tête, deux états à la même étape**

cohérence $\wedge$ **début**$_x$ $\wedge$ **accepte**

On est où ? 🧭

deux symboles ne sont pas assignés
à la même case, deux positions
à la même tête, deux états à la même étape

cohérence $\wedge$ **début**$_x$ $\wedge$ **accepte**

la configuration initiale
est la bonne

On est où ? 🧭

deux symboles ne sont pas assignés
à la même case, deux positions
à la même tête, deux états à la même étape

$$\textbf{cohérence} \wedge \textbf{début}_x \wedge \textbf{accepte}$$

la configuration initiale
est la bonne

on arrive dans un état
acceptant à un temps $\leq t$

Il reste le cœur de la simulation :
spécifier que le comportement
de la machine correspond
à la relation de transition

Sous-formule transition$_j$

$$\psi_{\textbf{contenu}} = \bigwedge_i \left(\neg p_{i,j-1} \rightarrow \bigwedge_\gamma \left(c_{\gamma,i,j} \leftrightarrow c_{\gamma,i,j-1} \right) \right)$$

Sous-formule transition$_j$

$$\psi_{\textbf{contenu}} = \bigwedge_i \left(\neg p_{i,j-1} \to \bigwedge_\gamma \left(c_{\gamma,i,j} \leftrightarrow c_{\gamma,i,j-1}\right)\right)$$

si la tête n'est pas sur
la case i à l'étape $j-1$...

Sous-formule transition$_j$

$$\psi_{\textbf{contenu}} = \bigwedge_i \left(\neg p_{i,j-1} \to \bigwedge_\gamma \left(c_{\gamma,i,j} \leftrightarrow c_{\gamma,i,j-1} \right) \right)$$

si la tête n'est pas sur la case i à l'étape $j-1$...

...alors le symbole sur cette case ne change pas à l'étape j

Sous-formule transition_j

$$\psi_{\textbf{trans}} = \bigwedge_{q,i,\gamma} \Big(\big(e_{q,j-1} \land p_{i,j-1} \land c_{\gamma,i,j-1} \big) \rightarrow \bigvee_{(q',\gamma',d') \in \delta(q,\gamma)} \big(e_{q',j} \land c_{\gamma',i,j} \land p_{i+d',j} \big) \Big)$$

rappel : i est une position, j une étape, γ un symbole, q un état, $d' \in \{-1, 0, +1\}$

Sous-formule transition$_j$

si la machine est dans l'état q

à l'étape $j-1$...

$$\psi_{\textbf{trans}} = \bigwedge_{q,i,\gamma} \Big(\big(e_{q,j-1} \wedge p_{i,j-1} \wedge c_{\gamma,i,j-1} \big) \to \bigvee_{(q',\gamma',d')\in\delta(q,\gamma)} \big(e_{q',j} \wedge c_{\gamma',i,j} \wedge p_{i+d',j} \big) \Big)$$

rappel : i est une position, j une étape, γ un symbole, q un état, $d' \in \{-1, 0, +1\}$

Sous-formule transition$_j$

si la machine est dans l'état q
à l'étape $j-1$...

...est que sa têtes est sur
la case i et lit γ...

$$\psi\textbf{trans} = \bigwedge_{q,i,\gamma} \Big(\big(e_{q,j-1} \wedge p_{i,j-1} \wedge c_{\gamma,i,j-1} \big) \to \bigvee_{(q',\gamma',d') \in \delta(q,\gamma)} \big(e_{q',j} \wedge c_{\gamma',i,j} \wedge p_{i+d',j} \big) \Big)$$

rappel : i est une position, j une étape, γ un symbole, q un état, $d' \in \{-1, 0, +1\}$

Sous-formule transition_j

si la machine est dans l'état q à l'étape $j-1$...

...est que sa têtes est sur la case i et lit γ...

$$\psi_{\textbf{trans}} = \bigwedge_{q,i,\gamma} \left(\left(e_{q,j-1} \wedge p_{i,j-1} \wedge c_{\gamma,i,j-1} \right) \rightarrow \bigvee_{(q',\gamma',d') \in \delta(q,\gamma)} \left(e_{q',j} \wedge c_{\gamma',i,j} \wedge p_{i+d',j} \right) \right)$$

...alors la machine fait l'une des transitions décrites par sa relation δ à l'étape j

rappel : i est une position, j une étape, γ un symbole, q un état, $d' \in \{-1, 0, +1\}$

Sous-formule transition_j

$$\textbf{transition}_j = \psi_{\textbf{contenu}} \wedge \psi_{\textbf{trans}}$$

Sous-formule transition_j

les cases qui ne sont pas sous la tête ne changent pas

$$\text{transition}_j = \psi_{\text{contenu}} \wedge \psi_{\text{trans}}$$

Sous-formule transition_j

les cases qui ne sont pas sous
la tête ne changent pas

$$\text{transition}_j = \psi_{\textbf{contenu}} \wedge \psi_{\textbf{trans}}$$

celles qui le sont changent selon
la relation de transition δ

Le pire est passé ! 🎊

Enfin, voilà la formule φ_x

$$\textbf{cohérence} \wedge \textbf{début}_x \wedge \bigwedge_{1 \leq j \leq t} \textbf{transition}_j \wedge \textbf{accepte}$$

Enfin, voilà la formule φ_x

deux valeurs ne sont pas assignés
à la même case, deux positions
à la même tête, deux états à la même étape

$$\textbf{cohérence} \wedge \textbf{début}_x \wedge \bigwedge_{1 \leq j \leq t} \textbf{transition}_j \wedge \textbf{accepte}$$

la configuration initiale
est la bonne

on faite une bonne
transition à chaque étape

on arrive dans un état
acceptant à un temps $\leq t$

Enfin, voilà la formule φ_x

$$\bigwedge_{r,i,j} \bigvee_{\gamma} \left(c^r_{\gamma,i,j} \wedge \bigwedge_{\gamma' \neq \gamma} \neg c^r_{\gamma,i,j} \right) \wedge \bigwedge_{r,j} \bigvee_{i} \left(p^r_{i,j} \wedge \bigwedge_{i' \neq i} \neg p^r_{i',j} \right) \wedge \bigwedge_{j} \bigvee_{q} \left(e_{q,j} \wedge \bigwedge_{q' \neq q} \neg e_{q',j} \right)$$

$$\wedge$$

$$e_{q_0,0} \wedge \bigwedge_{r>1,i} c^r_{B,i,0} \wedge \left(\bigwedge_{i \leq 0 \vee i > n} C^1_{B,i,0} \right) \wedge \left(\bigwedge_{1 \leq i \leq n} C^1_{x_i,i,0} \right) \wedge \bigwedge_{r} p^r_{1,0}$$

$$\wedge$$

$$= \bigwedge_{q,\vec{i},\vec{\gamma}} \left(\left(e_{q,j-1} \wedge \bigwedge_{r} \left(p^r_{i_r,j-1} \wedge c^r_{\gamma_r,i_r,j-1} \right) \right) \rightarrow \bigvee_{(q',\overrightarrow{\gamma'},\overrightarrow{d'}) \in \delta(q,\vec{\gamma})} \left(e_{q',j} \wedge \left(\bigwedge_{r} c^r_{\gamma'_r,i_r,j} \wedge p^r_{i_r+d'_r,j} \right) \right) \right)$$

$$\wedge$$

$$e_{q_a,t}$$

Enfin, voilà la formule φ_x

$$\bigwedge_{i,j}\bigvee_{\gamma}\left(c_{\gamma,i,j}\wedge\bigwedge_{\gamma'\neq\gamma}\neg c_{\gamma,i,j}\right)\wedge\bigwedge_{j}\bigvee_{i}\left(p_{i,j}\wedge\bigwedge_{i'\neq i}\neg p_{i',j}\right)\wedge\bigwedge_{j}\bigvee_{q}\left(e_{q,j}\wedge\bigwedge_{q'\neq q}\neg e_{q',j}\right)$$

$$\wedge$$

$$e_{q_0,0}\wedge\left(\bigwedge_{i\leq 0\vee i>n}c_{B,i,0}\right)\wedge\left(\bigwedge_{1\leq i\leq n}c_{x_i,i,0}\right)\wedge\bigwedge_{r}p_{1,0}$$

$$\wedge$$

$$=\bigwedge_{q,i,\gamma}\Big(\left(e_{q,j-1}\wedge p_{i_r,j-1}\wedge c_{\gamma_r,i_r,j-1}\right)\rightarrow\bigvee_{(q',\gamma',d')\in\delta(q,\vec{\gamma})}\left(e_{q',j}\wedge c_{\gamma'_r,i_r,j}\wedge p_{i_r+d'_r,j}\right)\Big)$$

$$\wedge$$

$$e_{q_a,t}$$

$|\varphi_x|$ = polynomial

par rapport à n

φ_x est satisfaisable
ssi N accepte x !

Sous-formule cohérence

$$\bigwedge_{i,j} \bigvee_{\gamma} \left(c_{\gamma,i,j} \wedge \bigwedge_{\gamma' \neq \gamma} \neg c_{\gamma',i,j} \right)$$

à tout moment chaque case contient exactement un symbole

$$\wedge$$

$$\bigwedge_{j} \bigvee_{i} \left(p_{i,j} \wedge \bigwedge_{i' \neq i} \neg p_{i',j} \right)$$

à tout moment la tête a exactement une position

$$\wedge$$

$$\bigwedge_{j} \bigvee_{q} \left(e_{q,j} \wedge \bigwedge_{q' \neq q} \neg e_{q',j} \right)$$

à tout moment on est dans exactement un état

rappel : i est une position, j une étape, γ un symbole

Sous-formule début_x

$$e_{q_0,0}$$

l'état de départ est l'état initial q_0

$$\wedge$$

$$\Big(\bigwedge_{i\leq 0 \vee i>n} c_{B,i,0}\Big) \wedge \Big(\bigwedge_{1\leq i\leq n} c_{x_i,i,0}\Big)$$

le ruban contient x entre les positions 1 et n et B ailleurs

$$\wedge$$

$$p_{1,0}$$

la tête de lecture est en position 1 au temps 0

rappel : i est une position, j une étape, γ un symbole

Sous-formule accepte

- Pour alléger les notations qui suivent et par abus de notation on pose

$$\delta(q_{oui}, \gamma) = \{(q_{oui}, \gamma, 0)\}$$
$$\delta(q_{no}, \gamma) = \{(q_{no}, \gamma, 0)\}$$

- C'est-à-dire, quand on accepte ou rejette, les configurations suivantes restent identiques

- Du coup on a tout simplement accepte $= e_{q_{oui},t}$
(au temps t on est dans l'état acceptant q_{oui})

Sous-formule transition_j

$$\psi_{\text{contenu}} = \bigwedge_{r,i} \left(\neg p^r_{i,j-1} \to \bigwedge_{\gamma} \left(c^r_{\gamma,i,j} \leftrightarrow c^r_{\gamma,i,j-1} \right) \right)$$

si la tête n'est pas sur la case i à l'étape $j-1$...

...alors le symbole sur cette case ne change pas à l'étape j

Sous-formule transition$_j$

si la machine est dans l'état q
à l'étape $j-1$…

…est que sa têtes est sur
la case i et lit γ…

$$\psi_{\textbf{trans}} = \bigwedge_{q,i,\gamma} \Big(\big(e_{q,j-1} \wedge p_{i,j-1} \wedge c_{\gamma,i,j-1} \big) \rightarrow \bigvee_{(q',\gamma',d') \in \delta(q,\vec{\gamma})} \big(e_{q',j} \wedge c_{\gamma',i,j} \wedge p_{i+d',j} \big) \Big)$$

…alors la machine fait l'une des transitions
décrites par sa relation δ à l'étape j

rappel : i est une position, j une étape, γ un symbole, q un état, $d' \in \{-1, 0, +1\}$

Conclusion de la démonstration

- On a pris $B \in \mathbf{NP}$ et une machine non déterministe N qui le reconnaît en temps polynomial $p(n)$

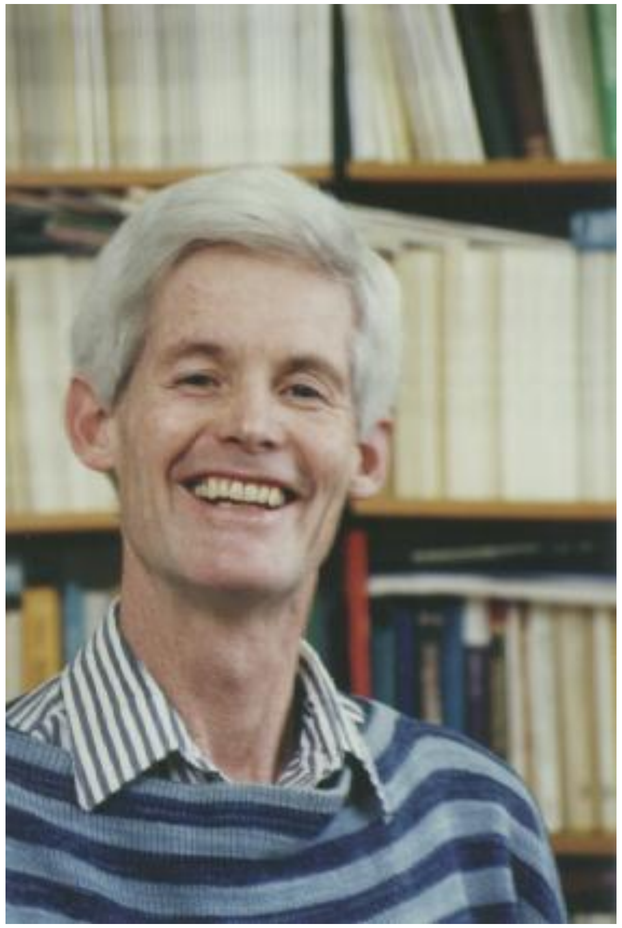
- Pour chaque entrée x de B on construit une formule φ_x de taille polynomiale qui décrit le calcul de $N(x)$

- Cette construction on peut la faire en temps polynomial, parce que la formule φ_x ne sera peut-être pas jolie, mais elle est régulière

- En plus on a $x \in B$ ssi $\varphi_x \in$ SAT

- Donc $B \leq$ SAT, et comme B était un langage quelconque dans $\mathbf{NP}$, on obtient la $\mathbf{NP}$-complétude de SAT

👍



Stephen Cook

Леони́д Ле́вин

SAT est $\mathbf{NP}$-complete