

# Chapitre 3.

## Modèles temporisés de systèmes à événements discrets

### 3.1 Introduction

Ce chapitre est consacré aux **modèles temporisés** des SED.

En général, lorsqu'un système combine une **dynamique classique, résultant de l'écoulement du temps, avec une dynamique engendrée par les événements**, cela donne ce qu'on appelle un **système hybride**.

L'instance la plus simple de dynamique induite par le temps combinée avec une dynamique liée aux événements consiste à adjoindre **une ou plusieurs horloges** à un modèle non-temporisé de SED, ce qui résulte en un type particulier de modèle hybride appelé **modèle temporisé**.

Dans les systèmes hybrides la dynamique induite par le temps est plus compliquée que celle des horloges. A chaque état discret du système est associé un système d'équations différentielles décrivant l'évolution des variables continues d'intérêt.

Dans les modèles de SED temporisés, les trajectoires ne sont plus spécifiées par des séquences d'événements  $(e_1, e_2, \dots)$  ou des séquences d'états  $(x_0, x_1, \dots)$ , mais doivent aussi inclure une forme d'information temporelle. Ainsi, si  $t_k$ ,  $k = 1, 2, \dots$ , est l'instant où le  $k$ ème événement et la transition d'état associée se produisent (avec  $t_0$  donné), alors une trajectoire temporisée d'un SED peut être décrite par la séquence  $((x_0, t_0), (x_1, t_1), \dots)$ . De même une séquence temporisée d'événements est de la forme  $((e_1, t_1), (e_2, t_2), \dots)$ .

Nous allons développer des **outils d'analyse des modèles temporisés de SED** qui vont nous permettre de répondre à des questions telles que

- « **Combien d'événements** d'un certain type peuvent se produire dans un intervalle de temps donné ? »,
- « Est-ce que **l'intervalle de temps** entre les occurrences de deux événements est toujours plus grand qu'une borne inférieure donnée ? », ou
- « **Combien de temps** le système passe-t-il dans un certain état ? »

Les réponses à ces questions sont essentielles lors de l'analyse du comportement de nombreuses classes de SED car elles fournissent des **mesures** particulièrement utiles **de la performance des systèmes**.

Le **premier objectif** de ce chapitre est de décrire les **mécanismes de temporisation** pour les SED. A cette fin, nous allons introduire la notion de **structure d'horloges**, qui capturera les contraintes temporelles associées aux occurrences consécutives de chaque (type d') événement dans le modèle et servira d'entrée au modèle non-temporisé.

On verra qu'en utilisant le mécanisme de structure d'horloges, il est possible d'**enrichir les automates et réseaux de Petri non-temporisés** introduits précédemment pour **créer des schémas de modélisation** pour les SED incluant le **chronométrage des événements**.

Le **second objectif** est d'établir le **cadre des automates temporisés et des réseaux de Petri temporisés avec des structures d'horloges** (Sections 3.2 et 3.3).

Il deviendra clair à la lecture de ce chapitre que toutes les informations requises en entrée des modèles temporisés sont censées être disponibles, soit sous forme d'une structure d'horloges, soit sous forme d'intervalles de temps.

Ceci signifie que nous disposons d'un moyen de décrire d'une manière **non-stochastique** quand des événements sont autorisés à se produire dans le futur.

Bien sûr, il est irréaliste de s'attendre à ce que ce soit toujours le cas. Un utilisateur de système informatique n'annonce pas les travaux qu'il soumettra et leurs dates de soumission pour la semaine prochaine. De même quelqu'un ne planifie pas des jours en avance quand il va placer divers appels téléphoniques.

En bref, pour effectuer des analyses d'utilité pratique, nous aurons besoin au bout du compte de faire appel aux **modèles probabilistes**; ce sera le propos du **chapitre 4**.

Nous nous limiterons dans ce chapitre à une description de la temporisation totalement libre d'aspects stochastiques, notre objectif étant de comprendre ce qui est fondamental pour l'évolution dynamique d'un SED lorsque la temporisation des événements est prise en compte.

## 3.2 Automates temporisés

Notre modèle de départ est un **automate**  $G = (\mathbf{X}, \mathbf{E}, f, \Gamma, x_0)$  où

$\mathbf{X}$  est un *espace d'états* dénombrable,

$\mathbf{E}$  est un *ensemble* dénombrable *d'événements*,

$f: \mathbf{X} \times \mathbf{E} \rightarrow \mathbf{X}$  est une *fonction de transition d'état* et est généralement une *fonction partielle* sur son domaine,

$\Gamma: \mathbf{X} \rightarrow 2^{\mathbf{E}}$  est la *fonction événements actifs* (ou *fonction événements possibles*),

$\Gamma(x)$  est l'ensemble de tous les événements  $e$  pour lesquels  $f(x, e)$  est définie et est appelé *l'ensemble d'événements actifs* (ou ensemble d'événements possibles),

$x_0$  est l'état initial.

C'est essentiellement l'automate défini au chapitre 2, avec des modifications mineures :

D'une part, les ensembles  $X$  et  $E$  sont dénombrables (plutôt que juste finis).

D'autre part, toute considération d'états marqués est laissée de côté de la définition car on ne se préoccupera pas de problèmes de blocage dans ce chapitre.

### *3.2.1 La structure d'horloges*

Afin d'introduire les idées clés du mécanisme de temporisation dont nous avons besoin, nous considérons tout d'abord le SED le plus simple possible. Nous progresserons ensuite graduellement avec des cas plus compliqués.

#### *Un SED avec un seul événement*

Pour ce système,  $E = \{a\}$  et  $\Gamma(x) = \{a\}$  pour tout  $x \in X$ . Une trajectoire sur la ligne du temps, ou chronogramme, est montrée dans la figure 3.1.

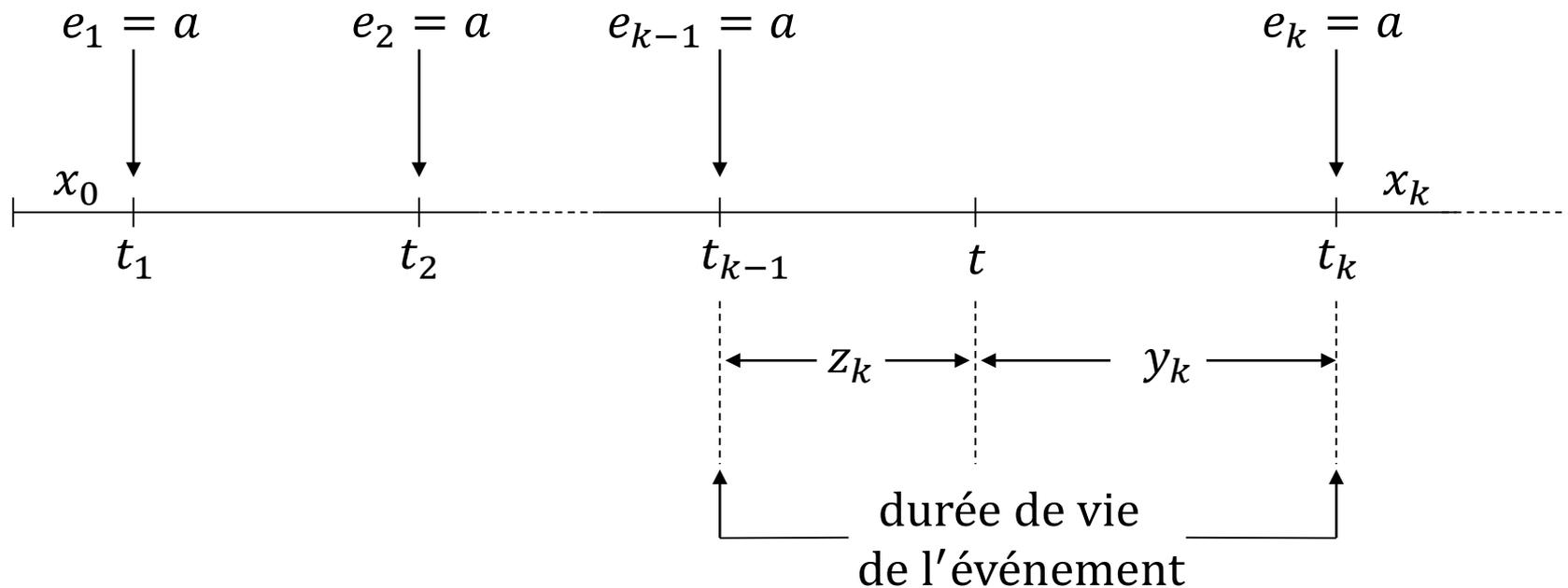
La séquence d'événements associée à cette trajectoire est notée  $(e_1, e_2, \dots)$ , où  $e_k = a$  pour tout  $k = 1, 2, \dots$

L'instant associé à la  $k$ ème occurrence est noté  $t_k$ ,  $k = 1, 2, \dots$

La longueur de l'intervalle de temps défini par deux occurrences successives de l'événement est appelée une **durée de vie**. Ainsi, on définit la  $k$ ème durée de vie de l'événement selon

$$v_k = t_k - t_{k-1}, \quad k = 1, 2, \dots \quad (3.1)$$

C'est un nombre réel non négatif :  $v_k \in \mathbb{R}^+$



**Figure 3.1.** Chronogramme d'un SED avec  $E = \{a\}$ .

L'évolution de ce système au cours du temps peut être décrite comme suit. A l'instant  $t_{k-1}$ , le  $k$ ème événement est dit être **activé** ou **permis**, et une durée de vie  $v_k$  lui est associée. Une horloge, ou une minuterie, associée à l'événement est immédiatement réglée à la valeur spécifiée par  $v_k$ , et commence alors à s'égrener jusqu'à 0. Pendant cet intervalle de temps, le  $k$ ème événement est dit **actif**. L'horloge atteint 0 quand la durée de vie expire, à l'instant  $t_k = t_{k-1} + v_k$ . A ce point, l'événement doit se produire. Ceci va causer une transition d'état. Le processus se répète alors, avec le  $(k + 1)$ ème événement devenant actif.

Il faut bien comprendre la différence entre un événement **devenant actif** et **se produisant**. L'événement  $a$  est actif tant qu'il est possible dans l'état courant, c'est-à-dire que  $a \in \Gamma(x)$ . L'événement se produit effectivement quand son minuteur atteint 0, et une transition d'état est alors effectuée. Cette distinction est similaire à celle vue au chapitre 2 entre **autoriser une transition** et **la franchir** dans les réseaux de Petri.

Dans la figure 3.1, l'événement  $a$  est en fait actif de façon permanente, mais il se produit seulement aux instants  $t_1, t_2, \dots$ . Chaque fois qu'il se produit, il est immédiatement activé à neuf, puisqu'il est toujours autorisé dans cet exemple.

Pour introduire des notations complémentaires, considérons un instant quelconque  $t$ , pas nécessairement associé à l'occurrence d'un événement. Supposons que  $t_{k-1} \leq t \leq t_k$ . Alors  $t$  divise l'intervalle  $[t_{k-1}, t_k]$  en deux parts (voir figure 3.1) telles que

$$y_k = t_k - t \quad (3.2)$$

est appelé l'**horloge** ou la **durée de vie résiduelle** du  $k$ ème événement, et

$$z_k = t - t_{k-1} \quad (3.3)$$

est appelé l'**âge** du  $k$ ème événement. Bien évidemment

$$v_k = z_k + y_k \quad (3.4)$$

Il devrait être clair qu'un chronogramme de ce SED est entièrement spécifié par la séquence de durées de vie  $(v_1, v_2, \dots)$ . Cette séquence est aussi appelée la **séquence d'horloges** de l'événement  $a$ .

### *Un SED avec deux événements actifs en permanence*

Les choses deviennent plus intéressantes quand on considère un SED avec  $E = \{a, b\}$ .

Pour faire simple, on suppose tout d'abord que  $\Gamma(x) = \{a, b\}$  pour tout  $x \in X$ , de sorte que les deux événements sont toujours actifs.

Supposons qu'une séquence d'horloges est spécifiée pour chaque événement, c.-à-d. que sont connues les deux séquences de durées de vies :

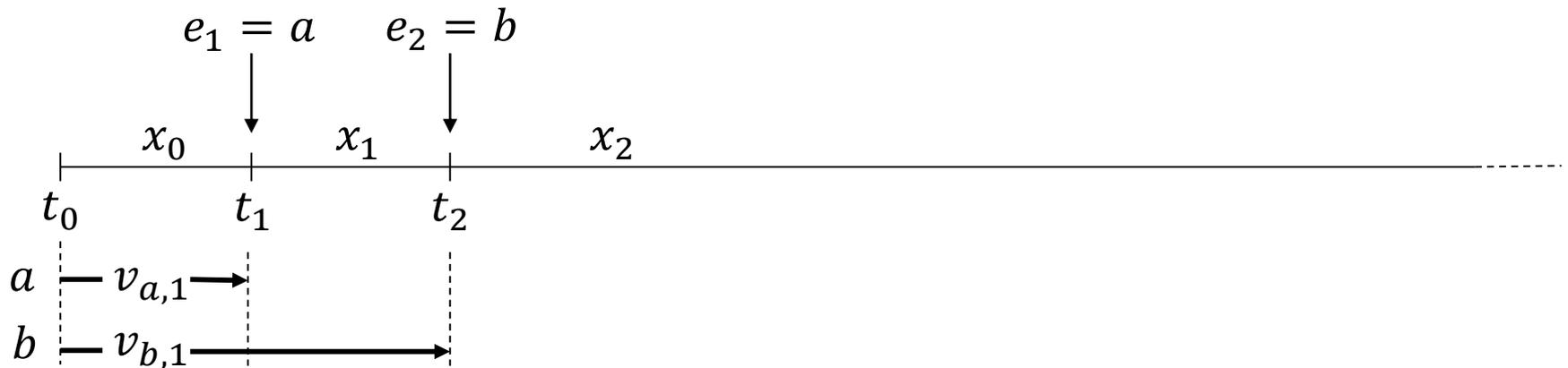
$$\mathbf{v}_a = (v_{a,1}, v_{a,2}, \dots) \text{ et } \mathbf{v}_b = (v_{b,1}, v_{b,2}, \dots).$$

Débutant à un instant donné  $t_0$ , la première question qui se pose est :

*Quel événement se produit ensuite ?*

Une réponse raisonnable à cette question est de comparer  $v_{a,1}$  et  $v_{b,1}$  et de sélectionner l'événement avec la durée de vie la plus courte.

Ainsi, comme l'illustre la figure 3.2 montrée en partie ci-dessous, si  $v_{a,1} < v_{b,1}$ ,  $a$  est le premier événement à se produire, à l'instant  $t_1 = t_0 + v_{a,1}$ .



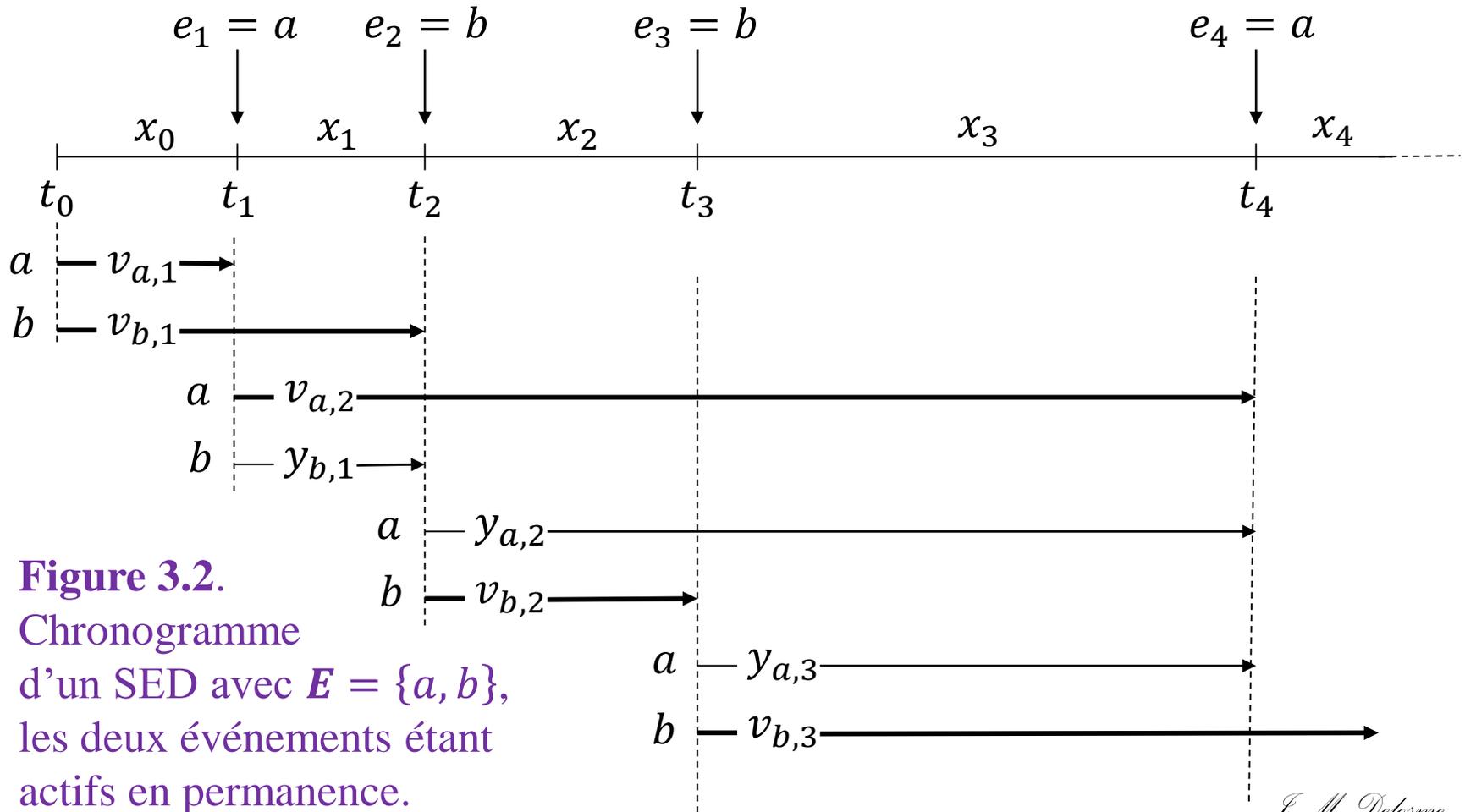
Nous sommes maintenant à l'instant  $t_1$  et pouvons demander à nouveau :  
*Quel événement se produit ensuite ?*

Comme  $b$  est déjà actif, sa valeur d'horloge est donnée par  $y_{b,1} = v_{b,1} - v_{a,1}$ .

D'autre part  $a$  est aussi actif.

Puisque  $a$  vient juste de se produire, sa prochaine occurrence est définie par la nouvelle durée de vie  $v_{a,2}$ , extraite de la séquence d'horloges  $\mathbf{v}_a$  donnée.

On compare maintenant  $v_{a,2}$  et  $y_{b,1}$  et sélectionne le plus petit des deux. Ainsi, si  $y_{b,1} < v_{a,2}$ ,  $b$  est le deuxième événement à se produire, à l'instant  $t_2 = t_1 + y_{b,1}$ , comme l'illustre la figure 3.2.



**Figure 3.2.**  
Chronogramme  
d'un SED avec  $E = \{a, b\}$ ,  
les deux événements étant  
actifs en permanence.

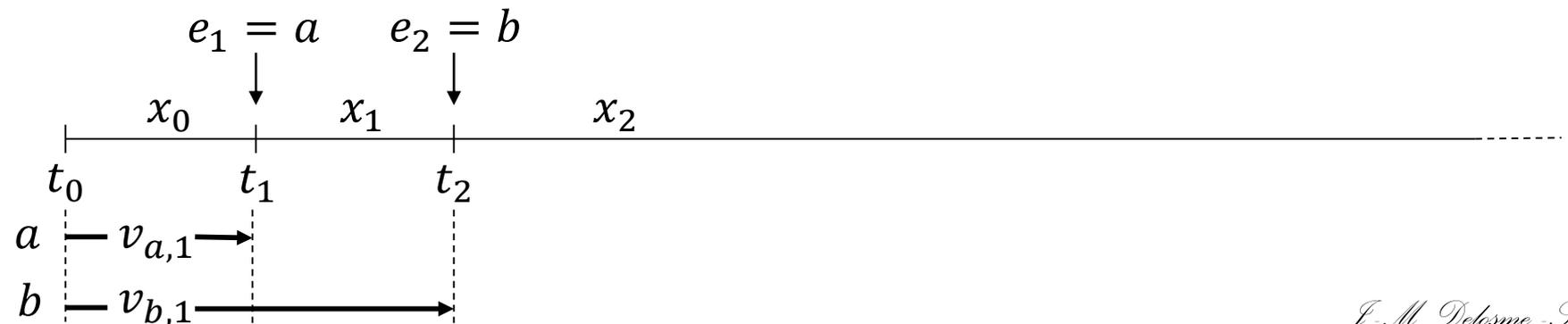
Le nouvel événement à se produire est toujours celui avec la plus petite valeur d'horloge. Quand un événement se produit, il est ici immédiatement activé derechef, et son horloge est réinitialisée à la valeur de durée de vie suivante dans la séquence d'horloges associée.

Le mécanisme pour sélectionner le « nouvel événement » à chaque instant est donc basé sur la comparaison des valeurs d'horloges et la sélection de la plus petite. Si un événement vient de se produire, alors sa minuterie est réglée sur une nouvelle valeur fournie par sa séquence de durées de vie.

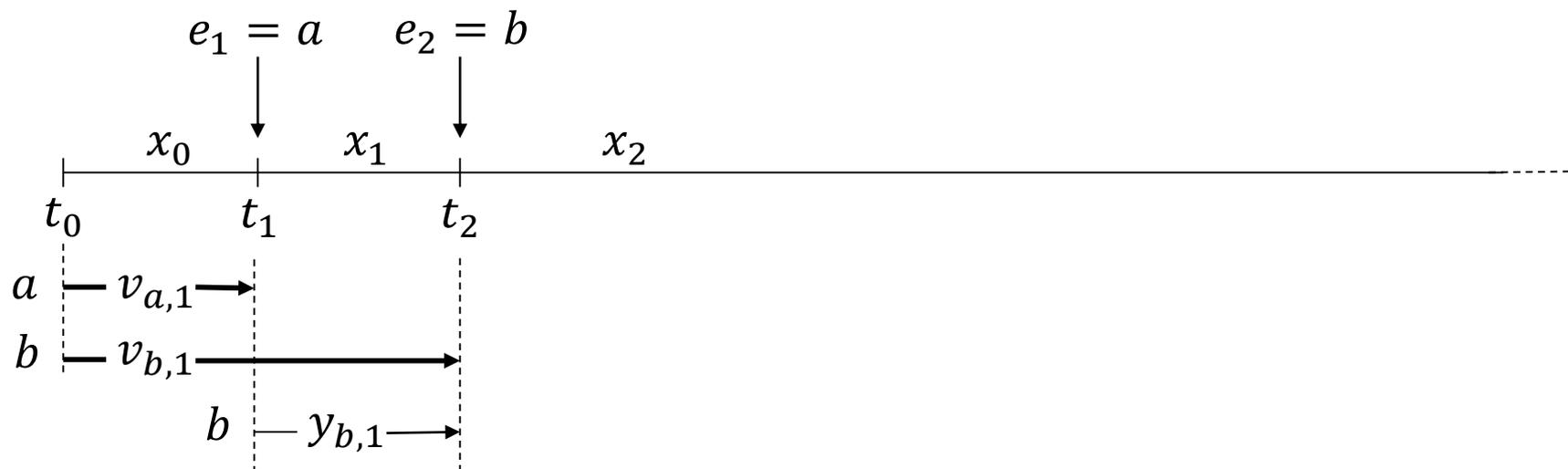
### *Un SED avec deux événements actifs par intermittence*

Notre schéma de modélisation ne serait pas très intéressant si aucun événement ne pouvait jamais être rendu inactif.

Supposons que dans le SED précédent,  $\Gamma(x) = \{a, b\}$  pour certains  $x \in X$ , et  $\Gamma(x) = \{b\}$  pour tous les autres  $x \in X$ . Si pour  $x_0$ , état initial donné,  $\Gamma(x_0) = \{a, b\}$ , alors, comme précédemment, le premier événement à se produire est  $a$  :

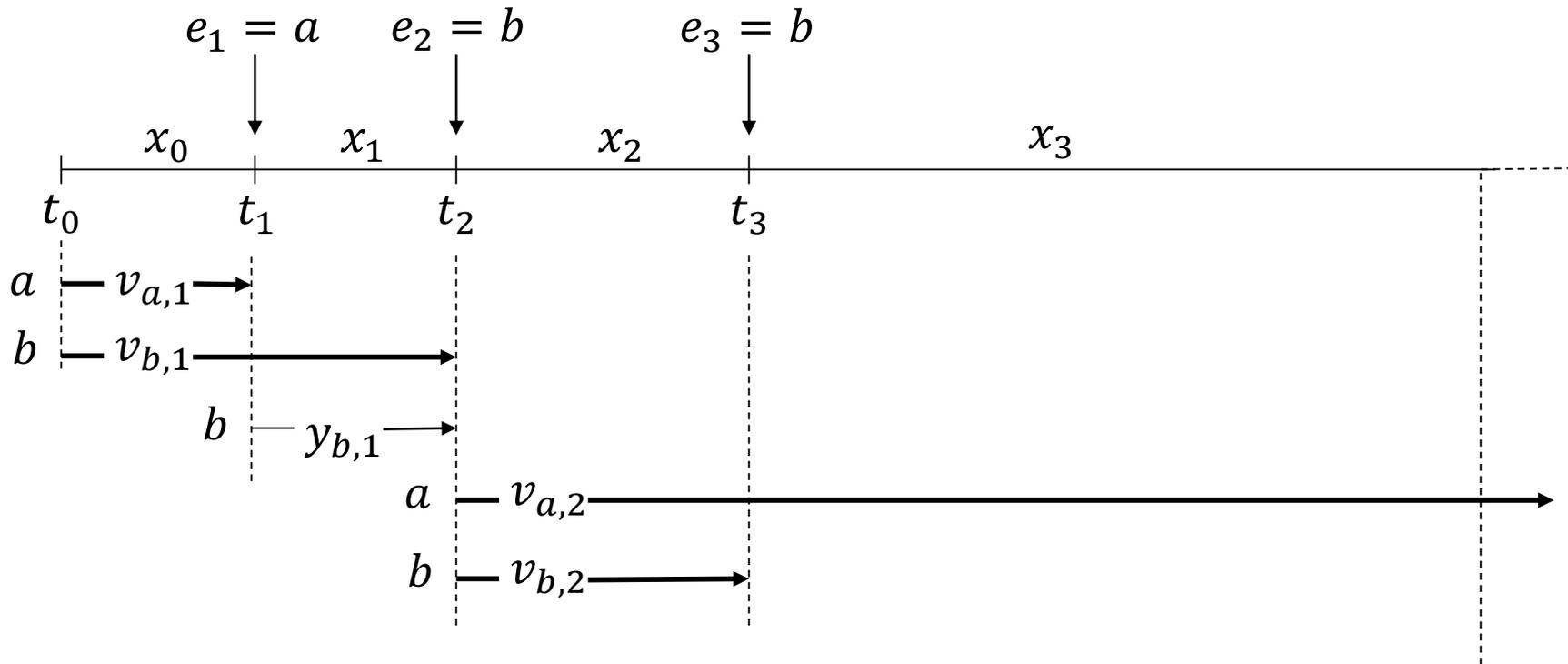


Supposons maintenant que le nouvel état,  $x_1$ , est tel que  $\Gamma(x_1) = \{b\}$ . Dans ce cas, le seul événement restant possible est  $b$ , et il n'y a donc pas besoin de comparaison avec aucune autre valeur d'horloge, cf. la portion de la figure 3.3 ci-dessous.



Ensuite, quand  $b$  se produit à l'instant  $t_2$ , il cause une transition dans l'état  $x_2$ .

Supposons qu'à nouveau  $a$  et  $b$  sont tous deux possibles. Dans ce cas, les deux événements sont activés. Comparant leurs valeurs d'horloges (les nouvelles durées de vies qui leur sont affectées à ce point), on trouve  $v_{b,2} < v_{a,2}$ , aussi est-ce l'événement  $b$  qui se produit ensuite (cf. la partie de la figure 3.3 donnée page suivante).

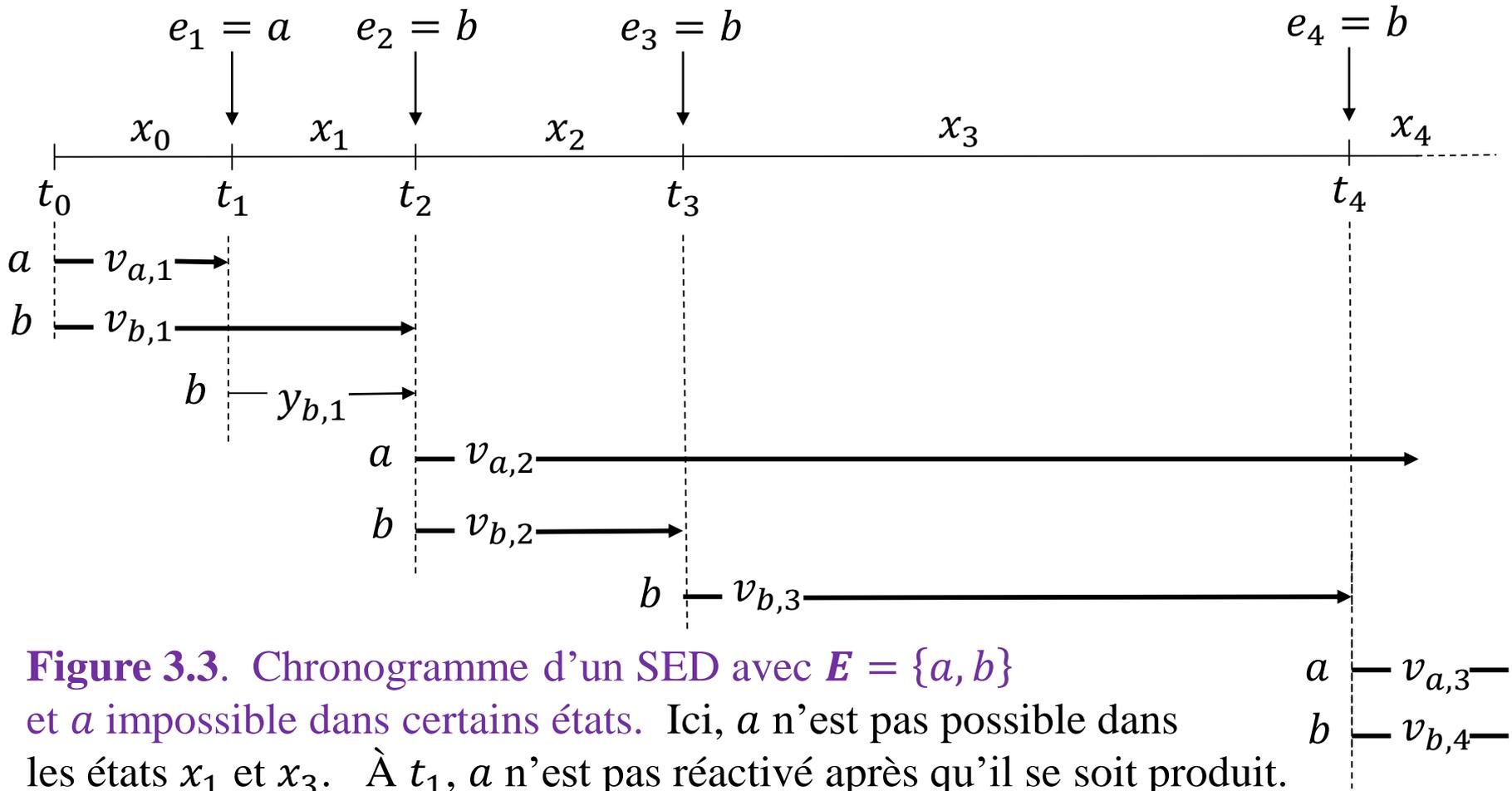


Supposons maintenant que le nouvel état,  $x_3$ , est tel que  $\Gamma(x_3) = \{b\}$ .  
 Puisque  $a$  n'est plus possible, il est **désactivé**. Son horloge n'importe plus pour déterminer l'événement suivant, et est donc ignorée.



Finalement, à l'instant  $t_4$ , advient  $b$ , forçant une transition vers un nouvel état  $x_4$ . Et, supposant  $\Gamma(x_4) = \{a, b\}$ , les deux événements sont activés encore une fois.

On remarque qu'une durée de vie entièrement nouvelle,  $v_{a,3}$ , est sélectionnée pour l'événement  $a$ , puisque  $v_{a,2}$  a été mis de côté lorsque  $a$  a été désactivé à l'instant  $t_3$ .



**Figure 3.3.** Chronogramme d'un SED avec  $E = \{a, b\}$

et  $a$  impossible dans certains états. Ici,  $a$  n'est pas possible dans les états  $x_1$  et  $x_3$ . À  $t_1$ ,  $a$  n'est pas réactivé après qu'il se soit produit. À  $t_3$ ,  $a$  est explicitement désactivé quand  $b$  se produit. La valeur d'horloge de  $a$  à ce point est mise au rebus. Quand  $a$  est réactivé à  $t_4$ , une nouvelle durée de vie est affectée à son horloge.

Ainsi, en général, le mécanisme pour sélectionner « l'événement suivant » à tout instant est basé sur trois règles simples :

**Règle 1.** Pour déterminer l'événement suivant, comparer les valeurs d'horloges de tous les événements possibles dans l'état courant et choisir la plus petite.

**Règle 2.** Un événement  $e$  est activé quand

- $e$  vient juste de se produire et il demeure possible dans le nouvel état (cf. par exemple, l'événement  $b$  à l'instant  $t_2$  dans la figure 3.3), ou
- un événement différent s'est produit alors que  $e$  n'était pas possible, causant une transition vers un nouvel état où  $e$  est possible (cf. l'événement  $a$  quand  $b$  se produit à l'instant  $t_2$  dans la figure 3.3).

**Règle 3.** Un événement  $e$  est désactivé quand un événement différent se produit causant une transition vers un nouvel état où  $e$  n'est pas possible (cf. l'événement  $a$  quand  $b$  se produit à l'instant  $t_3$  dans la figure 3.3).

Activer un événement est équivalent à régler son horloge à une nouvelle valeur de durée de vie extraite de la séquence (de valeurs) d'horloges qui lui est associée. Si un événement est désactivé, sa valeur d'horloge à ce moment est abandonnée.

Observons finalement que la détermination de l'événement suivant requiert

- (a) la connaissance de l'état courant  $x_k$ , de sorte que  $\Gamma(x)$  puisse être évalué, et
- (b) la connaissance de toutes les valeurs d'horloges courantes  $y_{i,k}$ , avec  $i \in \Gamma(x_k)$ .

Cette discussion, qui sera formalisée dans la prochaine section, fournit les principes fondamentaux pour le mécanisme de temporisation d'événements que nous utiliserons. La nouvelle notion que nous avons introduite ici, et qui sert à définir un modèle temporisé d'un SED, est une **structure d'horloges** :

**Définition :** La *structure d'horloges* (ou *structure de durées de vies*) associée à un ensemble d'événements  $E$  est un ensemble

$$V = \{v_i : i \in E\}$$

de séquences d'horloges (ou de durées de vies)

$$v_i = (v_{i,1}, v_{i,2}, \dots), \quad i \in E, \quad v_{i,k} \in \mathbb{R}^+, \quad k = 1, 2, \dots$$

La structure d'horloges sera considérée comme **l'entrée d'un SED**.

Le mécanisme de temporisation traduit alors cette information en une séquence effective d'événements qui dirige l'automate d'état sous-jacent.

On supposera que  $V$  est complètement spécifiée à l'avance.

Dans certains cas, une durée de vie  $v_{i,k}$  peut en fait devoir être transformée en fonction de l'état courant avant d'être affectée à l'horloge de l'événement  $i$ , mais ceci ne fait qu'ajouter des complications techniques qui n'affectent pas les principes des modèles.

Un autre concept utile est celui de **compte d'événements** :

**Définition** : Le **compte**,  $N_{i,k}$ , d'un événement  $i \in E$  après la  $k$ ème transition d'état sur une trajectoire donnée est le nombre de fois que  $i$  a été activé dans l'intervalle  $[t_0, t_k]$ .

Par exemple, dans la figure 3.2 les comptes sont initialisés à  $N_{a,0} = N_{b,0} = 1$ , puisque les deux événements sont activés à l'instant  $t_0$ . À l'instant  $t_1$  on a  $N_{a,1} = 2$  et  $N_{b,1} = 1$ . À l'instant  $t_2$  on a  $N_{a,2} = N_{b,2} = 2$ .

Ainsi, à chaque occurrence d'un événement, nous devons vérifier quelles sont les nouvelles activations et incrémenter de 1 les comptes d'événements correspondants. Le compte de l'événement  $i$  sert de pointeur à sa séquence d'horloges  $\mathbf{v}_i$ , qui spécifie la prochaine durée de vie à affecter à son horloge quand  $i$  est activé.

### 3.2.2 Dynamique temporelle des événements

Nous considérons un automate  $G = (\mathbf{X}, \mathbf{E}, f, \Gamma, x_0)$ .

Pour simplifier la présentation nous supposons que l'ensemble d'événements  $\mathbf{E}$  est fini et comprend  $m$  éléments.

Nous avons vu que la dynamique séquentielle du système est capturée par l'équation de transition d'état

$$x_{k+1} = f(x_k, e_{k+1}), \quad k = 0, 1, \dots \quad (3.5)$$

où  $x_k$  est l'état courant et  $x_{k+1}$  est le nouvel état obtenu quand l'événement  $e_{k+1}$  se produit.

Dans ce cadre, une séquence donnée d'événements  $(e_1, e_2, \dots, e_{k+1}, \dots)$  constitue l'entrée du système. Se rappelant la définition de « l'état », nous voyons que la connaissance de cette séquence (d'événements) d'entrée avec celle de  $x_k$  sont suffisantes pour prédire toute l'évolution future de l'état au moyen de (3.5).

Équipons maintenant l'automate  $G$  avec une structure d'horloges  $V = \{\mathbf{v}_i : i = 1, \dots, m\}$ . Cette structure d'horloges doit être considérée comme l'entrée de l'automate. La séquence d'événements  $(e_1, e_2, \dots, e_{k+1}, \dots)$  n'est alors plus donnée. En fait, elle doit être déterminée en utilisant un mécanisme de temporisation que nous définirons explicitement.

Nous cherchons donc maintenant une relation de la forme

$$e_{k+1} = h(x_k, \mathbf{v}_1, \dots, \mathbf{v}_m)$$

afin de remplacer (3.5) par une équation d'état

$$x_{k+1} = f(x_k, \mathbf{v}_1, \dots, \mathbf{v}_m)$$

Ceci semble impliquer que  $x_k$  demeure la seule variable d'état requise pour décrire le comportement dynamique d'un SED. C'est trop demander.

La discussion de la section précédente suggère plutôt que déterminer l'événement suivant,  $e_{k+1}$ , met en jeu un mécanisme de comparaisons d'horloges et de mises à jour, ainsi que le suivi des comptes d'événements.

Ainsi, notre objectif devient de spécifier le nouvel événement à l'aide d'un mécanisme de temporisation interne capturé par une relation de la forme

$$e_{k+1} = h(x_k, \mathbf{v}_1, \dots, \mathbf{v}_m, \cdot) \quad (3.6)$$

où «  $\cdot$  » représente des variables d'état additionnelles non encore déterminées.

Une fois cette relation totalement spécifiée, (3.5) et (3.6) prises ensemble décriront la dynamique de notre système.

Nous allons maintenant présenter en détail les mécanismes de temporisation d'événements décrits de manière informelle dans la section précédente. Pour faciliter la lecture, l'index de comptage d'événements  $k$  sera omis, et la notation suivante sera adoptée :

$x$  est l'état courant

$e$  est l'événement le plus récent (causant la transition dans l'état  $x$ )

$t$  est la date de l'événement le plus récent (correspondant à  $e$ )

En outre, deux variables sont associées à chaque événement  $i \in E$ :

$N_i$  est le **compte** courant de l'événement  $i$ ,  $N_i \in \{0, 1, \dots\}$

$y_i$  est la valeur courante de l'**horloge** de l'événement  $i$ ,  $y_i \in \mathbb{R}^+$

Les définitions de  $N_i$  et  $y_i$  ont été données informellement plus tôt; elles seront rendues mathématiquement précises dans ce qui suit.

L'état suivant, l'événement suivant, et la date de l'événement suivant sont représentés avec la notation prime ( $'$ ). Ainsi

$e'$  est l'événement suivant, appelé aussi l'**événement déclencheur**;  
clairement  $e' \in \Gamma(x)$

$t'$  est la date d'événement suivante (correspondant à l'événement  $e'$ )

$x'$  est l'état suivant, donné par  $x' = f(x, e')$

Similairement, on pose :

$N_i'$  le nouveau compte de l'événement  $i$  (après l'occurrence de  $e'$ )

$y_i'$  la nouvelle valeur d'horloge de l'événement  $i$  (après l'occurrence de  $e'$ )

**Remarque :** avec cette notation  $x$  correspond à  $x_k$ , et  $x'$  à  $x_{k+1}$ ,  $k = 0, 1, \dots$  et il en est de même pour toutes les autres variables définies plus haut.

Étant donné  $x$ , notre première tâche est de déterminer l'événement déclencheur  $e'$ . Nous le faisons de la manière suivante :

**Étape 1.** Puisque  $x$  est connu, on peut évaluer l'ensemble des événements possibles  $\Gamma(x)$ .

**Étape 2.** Une valeur d'horloge  $y_i$  est associée à chaque événement  $i \in E$ . On peut alors déterminer la plus petite valeur d'horloge parmi celles-ci, notée  $y^*$  :

$$y^* = \min_{i \in \Gamma(x)} \{y_i\} \quad (3.7)$$

**Étape 3.** Déterminer l'événement déclencheur,  $e'$ , comme la valeur de  $i$  dans (3.7) qui définit  $y^*$ . On l'exprime selon

$$e' = \arg \min_{i \in \Gamma(x)} \{y_i\} \quad (3.8)$$

**Étape 4.** Avec  $e'$  défini par (3.8), déterminer l'état suivant :

$$x' = f(x, e') \quad (3.9)$$

**Étape 5.** Avec  $y^*$  défini par (3.7), déterminer la date d'événement suivante :

$$t' = t + y^* \quad (3.10)$$

Le processus se répète alors avec  $x'$ ,  $e'$ , et  $t'$  spécifiés. L'étape 2, cependant, requiert les nouvelles valeurs d'horloge  $y'_i$ . Il faut donc encore au moins une autre étape :

**Étape 6.** Déterminer les nouvelles valeurs d'horloges pour tous les nouveaux événements possibles  $i \in \Gamma(x')$  :

$$y'_i = \begin{cases} y_i - y^* & \text{si } i \neq e' \text{ et } i \in \Gamma(x) \\ v_{i, N_i+1} & \text{si } i = e' \text{ ou } i \notin \Gamma(x) \end{cases} \quad i \in \Gamma(x') \quad (3.11)$$

Pour expliquer les deux cas dans (3.11), on observe que :

- (a) Le premier cas couvre tous les événements qui demeurent actifs dans l'état suivant  $x'$  à l'exception de l'événement déclencheur. Leurs valeurs d'horloge sont diminuées de  $y^*$  (la quantité de temps écoulée depuis l'occurrence de  $e$ ).
- (b) Le second cas traite l'événement déclencheur  $e'$  et tous les événements qui étaient inactifs dans l'état  $x$ , mais sont activés dans l'état suivant  $x'$  (c'est la Règle 2 de la section précédente). Tous ces événements requièrent une nouvelle durée de vie, qui est la suivante disponible dans la séquence d'horloges associée. Pour l'événement  $i$ , les  $N_i$  premières durées de vie ont déjà été utilisées, aussi la première suivante disponible est le  $(N_i+1)$ -ème élément dans la séquence  $\mathbf{v}_i$ .
- (c) On remarque que  $y'_i$  n'est pas défini pour tout  $i \notin \Gamma(x')$ . Ainsi, si  $i$  était actif dans l'état  $x$  et est désactivé quand arrive  $e'$ , sa valeur d'horloge est écartée (Règle 3 de la section précédente).

Quand le processus se répète, l'étape 6 fait appel à de nouvelles valeurs des comptes,  $N'_i$ . Il faut donc une dernière étape :

**Étape 7.** Déterminer les nouveaux comptes pour tous les nouveaux événements possibles  $i \in \Gamma(x')$  :

$$N'_i = \begin{cases} N_i + 1 & \text{si } i = e' \text{ ou } i \notin \Gamma(x) \\ N_i & \text{sinon} \end{cases} \quad i \in \Gamma(x') \quad (3.12)$$

Ici, le compte de l'événement  $i$  est incrémenté de 1 chaque fois que l'événement est activé. Cela arrive quand  $i \in \Gamma(x')$  et  $i$  est, soit l'événement déclencheur, soit un événement activé lors de l'entrée dans le nouvel état  $x'$ . Cette condition est identique au second cas de (3.11). La **plus petite valeur d'horloge d'événement,  $y^*$** , déterminée au moyen de (3.7), spécifie aussi la longueur de l'intervalle de temps défini par deux événements successifs. Ainsi on appellera  **$y^*$  la durée entre événements**.

**Remarque :** Il se pourrait que  $y_i = y_j$  pour deux événements distincts  $i, j \in \Gamma(x)$ . Pour résoudre ce conflit, nous devons imposer des règles de priorité sur l'ensemble d'événements  $E$ , c.-à-d., si deux événements se produisent simultanément, lequel doit être considéré comme affectant l'état en premier ? Si ce n'est pas spécifié autrement, on supposera que les événements sont classés par priorité, c.-à-d., l'événement  $i$  a une priorité supérieure à tout événement  $j > i$ .

Nous pouvons maintenant donner une définition formelle d'un automate temporisé.

**Définition :** Un *automate temporisé*,  $G_t$ , est un sextuplet

$$G_t = (X, E, f, \Gamma, x_0, V)$$

où  $V = \{v_i : i \in E\}$  est une structure d'horloges et  $(X, E, f, \Gamma, x_0)$  est un automate. L'automate fournit une séquence d'états  $x' = f(x, e')$  générée par une séquence d'événements  $(e_1, e_2, \dots)$  selon

$$e' = \arg \min_{i \in \Gamma(x)} \{y_i\}$$

avec les *valeurs d'horloges*  $y_i$ ,  $i \in E$ , définies par

$$y'_i = \begin{cases} y_i - y^* & \text{si } i \neq e' \text{ et } i \in \Gamma(x) \\ v_{i, N_i+1} & \text{si } i = e' \text{ ou } i \notin \Gamma(x) \end{cases} \quad i \in \Gamma(x')$$

où la *durée entre événements*  $y^*$  est définie selon

$$y^* = \min_{i \in \Gamma(x)} \{y_i\}$$

et les comptes d'événements  $N_i$ ,  $i \in E$ , sont définis par

$$N'_i = \begin{cases} N_i + 1 & \text{si } i = e' \text{ ou } i \notin \Gamma(x) \\ N_i & \text{sinon} \end{cases} \quad i \in \Gamma(x')$$

De plus, les conditions initiales sont :  $y_i = v_{i,1}$  et  $N_i = 1$  pour tout  $i \in \Gamma(x_0)$ .

Si  $i \notin \Gamma(x_0)$ , alors  $y_i$  n'est pas défini et  $N_i = 0$ .

### 3.2.3. Un modèle d'espace d'état

Notre objectif est ici de mettre l'accent sur le fait que la définition d'un automate temporisé est essentiellement la description d'un modèle d'espace d'état pour un système dynamique.

Commençant avec le mécanisme de transition d'état d'automate décrit par (3.9) :

$$x' = f(x, e')$$

on voit que l'on a besoin de spécifier  $e'$  à l'aide de (3.8) :

$$e' = \arg \min_{i \in \Gamma(x)} \{y_i\}$$

Comme  $y_i$  est défini au moyen de l'équation récursive (3.11), nous devons l'adopter comme une variable d'état de notre modèle. En outre, (3.11) dépend de  $N_i$ , qui est aussi défini récursivement selon (3.12), et est donc une variable d'état.

On reconnaît à ce point un modèle d'espace d'état complet où  $x, y_1, \dots, y_m, N_1, \dots, N_m$  sont les **variables d'état** et  $\mathbf{v}_1, \dots, \mathbf{v}_m$  sont les **séquences d'entrée** du système.

Pour obtenir une description plus compacte du modèle, définissons :

$f^x(\cdot)$  : la fonction « état suivant » dans (3.9)

$f_i^y(\cdot)$  : la  $i$ ème fonction « valeur d'horloge suivante » dans (3.11),  $i \in \Gamma(x')$

$f_i^N(\cdot)$  : la  $i$ ème fonction « valeur de compte suivante » dans (3.12),  $i \in \Gamma(x')$

et notons  $\mathbf{y}$  et  $\mathbf{N}$  les vecteurs d'horloges et comptes courants

$$\mathbf{y} = [y_1, \dots, y_m], \quad \mathbf{N} = [N_1, \dots, N_m]$$

Les variables d'état  $N_1, \dots, N_m$  sont seulement utilisées pour se mettre à jour elles-mêmes et aider à compter les occurrences d'événements. Elles n'affectent pas l'état  $x$  de l'automate ni les valeurs d'horloges  $y_1, \dots, y_m$ .

Quoique nous les gardions encore ici en tant que part de notre description, nous verrons plus tard qu'elles sont souvent superflues.

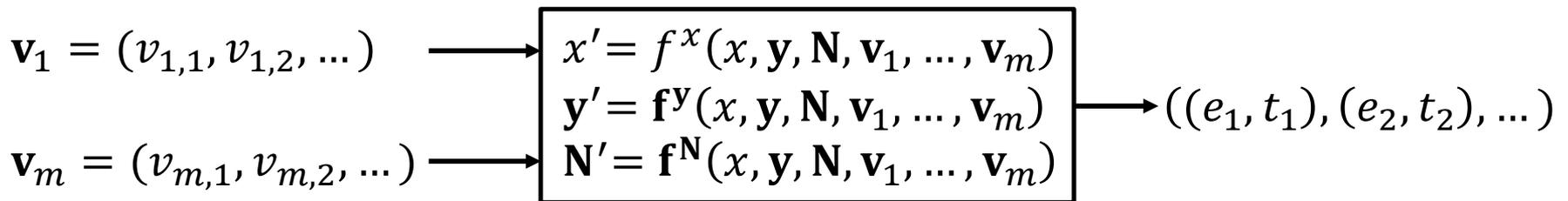
Nous pouvons maintenant écrire les équations d'état pour notre modèle selon :

$$x' = f^x(x, y, \mathbf{N}, \mathbf{v}_1, \dots, \mathbf{v}_m) \quad (3.13)$$

$$y' = \mathbf{f}^y(x, y, \mathbf{N}, \mathbf{v}_1, \dots, \mathbf{v}_m) \quad (3.14)$$

$$\mathbf{N}' = \mathbf{f}^N(x, y, \mathbf{N}, \mathbf{v}_1, \dots, \mathbf{v}_m) \quad (3.15)$$

Le modèle est résumé dans la figure 3.4.



**Figure 3.4.** Modèle d'état temporisé d'un SED : Les trois équations d'état sont données par (3.9), (3.11) et (3.12), et le temps est mis à jour selon  $t' = t + y^*$  où  $y^*$  est donné par (3.7).

L'entrée consiste en la structure d'horloges  $V = \{v_i: i = 1, \dots, m\}$ . La sortie est vue comme une séquence d'événements temporisés, qui définit la trajectoire dans le temps, ou le chronogramme, du SED.

Bien que la variable « temps », mise à jour avec (3.10), ne soit pas une part essentielle de la dynamique du modèle, elle est néanmoins pratique pour définir la séquence de sortie  $((e_1, t_1), (e_2, t_2), \dots)$ .

Les conditions initiales pour le modèle sont comme spécifié dans la définition d'un automate temporisé dans la section précédente. L'état initial  $x_0$  étant donné, on fixe  $N_{i,0} = 1$  pour tous les événements  $i \in \Gamma(x_0)$ , actifs initialement, et  $N_{i,0} = 0$  pour tout  $i \notin \Gamma(x_0)$ . On fixe aussi  $y_{i,0} = v_{i,1}$  pour tout  $i \in \Gamma(x_0)$ ,  $y_{i,0}$  n'est pas défini pour tout  $i \notin \Gamma(x_0)$ .

**Remarque :** D'un point de vue strict, l'état du système se compose des variables  $(x, y_1, \dots, y_m, N_1, \dots, N_m)$ . Toutes ces variables, accompagnées de la structure d'horloges  $V$ , sont nécessaires pour prédire le comportement futur du SED.

D'autre part, nous avons aussi parlé d'« état » au sujet de  $x$ , alors que  $x$  pris par lui-même ne décrit que partiellement le SED.

L'état  $x$  de l'automate sous-jacent est parfois appelé l'état physique ou l'état externe du système, en contraste avec les variables  $(y_1, \dots, y_m, N_1, \dots, N_m)$ , qui en constituent l'état interne.

Quand il est nécessaire de bien faire la distinction, nous appellerons  $x$  l'état du processus et  $(x, y_1, \dots, y_m, N_1, \dots, N_m)$  l'état du système.

En pratique on dira juste « état » pour  $x$  tout en gardant à l'esprit que l'état du système correct mathématiquement est défini par l'état  $x$  du processus accompagné des vecteurs de valeurs d'horloges et de comptes  $\mathbf{y}$  et  $\mathbf{N}$ .

Un modèle d'espace d'état alternatif utilise les âges des événements  $z_i, i = 1, \dots, m$ , au lieu des horloges  $y_i, i = 1, \dots, m$ . Ce modèle s'obtient facilement en utilisant (3.4). L'équation d'état (3.11) est remplacée par

$$z'_i = \begin{cases} z_i + y^* & \text{si } i \neq e' \text{ et } i \in \Gamma(x) \\ 0 & \text{si } i = e' \text{ ou } i \notin \Gamma(x) \end{cases} \quad i \in \Gamma(x') \quad (3.16)$$

où

$$y^* = \min_{i \in \Gamma(x)} \{(v_{i, N_i} - z_i)\} \quad (3.17)$$

$$e' = \arg \min_{i \in \Gamma(x)} \{(v_{i, N_i} - z_i)\} \quad (3.18)$$

et les équations d'état restantes (3.9) et (3.12) sont inchangées.

Dans (3.16), l'âge de tous les événements restant actifs dans le nouvel état  $x'$  est augmenté, à l'exception de l'événement déclencheur  $e'$ , et l'âge de tous les événements nouvellement activés est fixé à 0, incluant l'événement déclencheur, à condition que  $e' \in \Gamma(x')$ .

Les conditions initiales pour ce modèle sont aussi facilement spécifiées une fois que  $x_0$  est donné.

Comme précédemment, on fixe  $N_{i,0} = 1$  pour tous les événements actifs initialement  $i \in \Gamma(x_0)$ , et  $N_{i,0} = 0$  pour tout  $i \notin \Gamma(x_0)$ . On fixe de plus  $z_{i,0} = 0$  pour tout  $i \in \Gamma(x_0)$ , et  $z_{i,0}$  n'est pas défini pour tout  $i \notin \Gamma(x_0)$ .

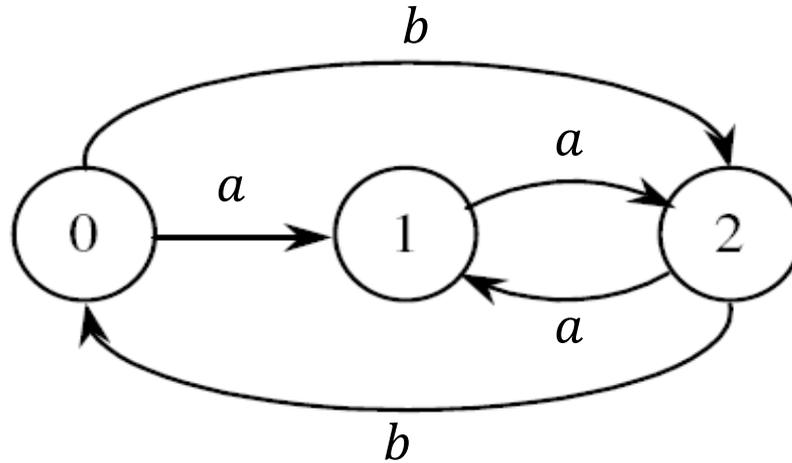
La complexité notationnelle du modèle d'espace d'état est principalement due à la nature discontinue des équations d'état pour les horloges et les comptes. Cependant, le modèle reflète un modèle intuitif simple pour générer des chronogrammes de SED, comme décrit initialement dans la section 3.2.1.

Laissant de côté la question de sa représentation mathématique, on observe que la génération effective de la trajectoire dans le temps s'appuie sur de simples comparaisons de nombres réels (pour déterminer la valeur d'horloge la plus petite, et donc l'événement déclencheur), et de l'arithmétique élémentaire (additions et soustractions) pour mettre à jour horloges et comptes d'événements.

### *Exemple 3.1*

On considère un automate avec  $E = \{a, b\}$  et  $X = \{0, 1, 2\}$ . Un diagramme de transition d'état est donné dans la figure 3.5. On voit que  $\Gamma(0) = \Gamma(2) = \{a, b\}$ , tandis que  $\Gamma(1) = \{a\}$ .

On suppose que l'état initial est  $x_0 = 0$  à  $t_0 = 0$ .



**Figure 3.5.** Diagramme de transition d'état pour l'automate temporisé de l'exemple 3.1

Une structure d'horloges est fournie comme suit (seules quelques durées de vies sont montrées) :

$$\mathbf{v}_a = (1, 1,5, 1,5, \dots), \quad \mathbf{v}_b = (2, 0,5, 1,5, \dots),$$

La trajectoire de ce SED temporisé pour les quatre premières occurrences d'événements est montrée dans la figure 3.6.

Initialement, les deux événements sont actifs et leurs valeurs d'horloges et de comptes sont

$$\begin{aligned}
 y_{a,0} = v_{a,1} = 1, & & y_{b,0} = v_{b,1} = 2 \\
 N_{a,0} = 1, & & N_{b,0} = 1
 \end{aligned}$$

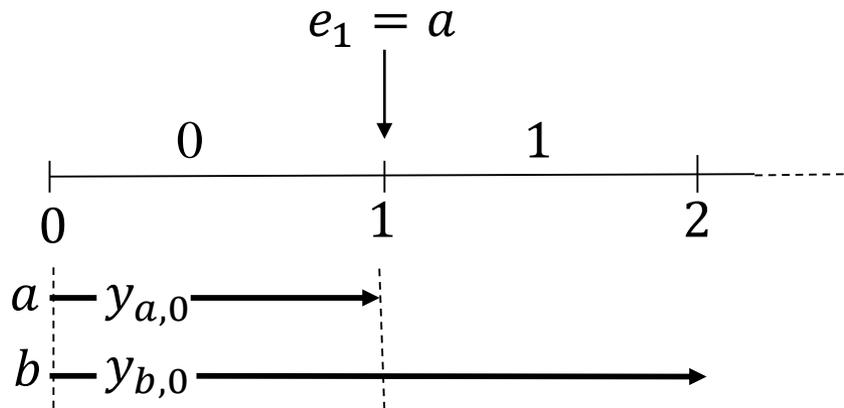
On peut maintenant déterminer la trajectoire entière par inspection, ce qui montre la facilité de reconstruction et d'application des équations du modèle, (3.7) à (3.12). D'abord,

$$y_0^* = \min \{y_{a,0}, y_{b,0}\} = y_{a,0} \Rightarrow e_1 = a$$

Donc, au vu du diagramme de transition d'état, l'état suivant est

$$x_1 = f^x(0, a) = 1$$

Aussi, la date de l'événement est simplement donnée par  $t_1 = y_0^* = 1$ .

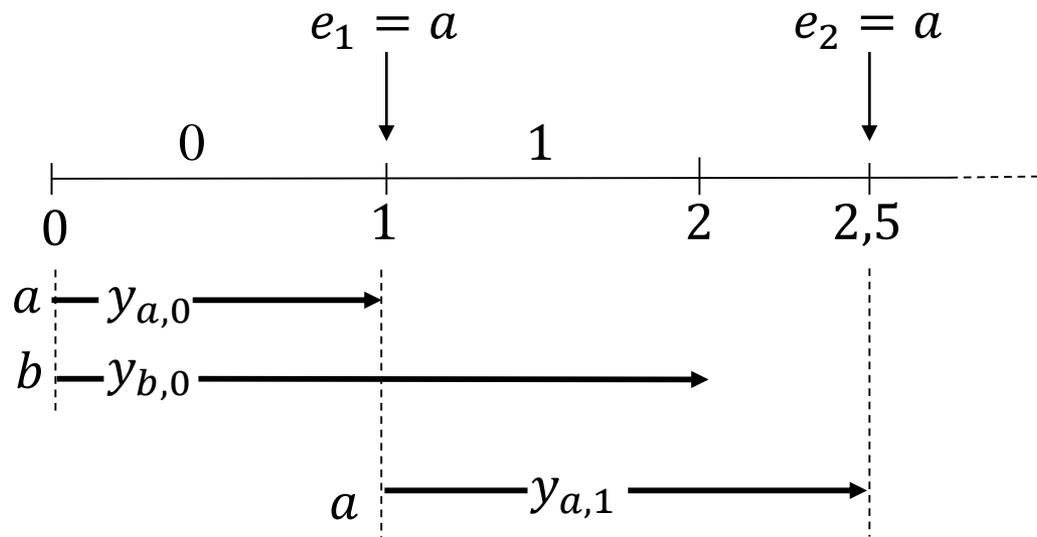


Répetons le processus avec  $\Gamma(1) = \{a\}$ .

On remarque que dans cet état la valeur d'horloge de  $b$  n'est plus pertinente.

Puisque  $a$  était l'événement déclencheur, il est activé à nouveau avec un compte incrémenté de 1. Sa nouvelle valeur de l'horloge est

$$y_{a,1} = v_{a,2} = 1,5$$



Évidemment, l'événement déclencheur est  $e_2 = a$ . Donc l'état suivant est

$$x_2 = f^x(1, a) = 2$$

et le temps avance jusqu'à  $t_2 = t_1 + y_1^* = 2,5$ .

Ensuite, on a  $\Gamma(2) = \{a, b\}$ , et les deux événements sont activés de nouveau.

Leurs comptes sont aussi incrémentés. Ainsi, on assigne

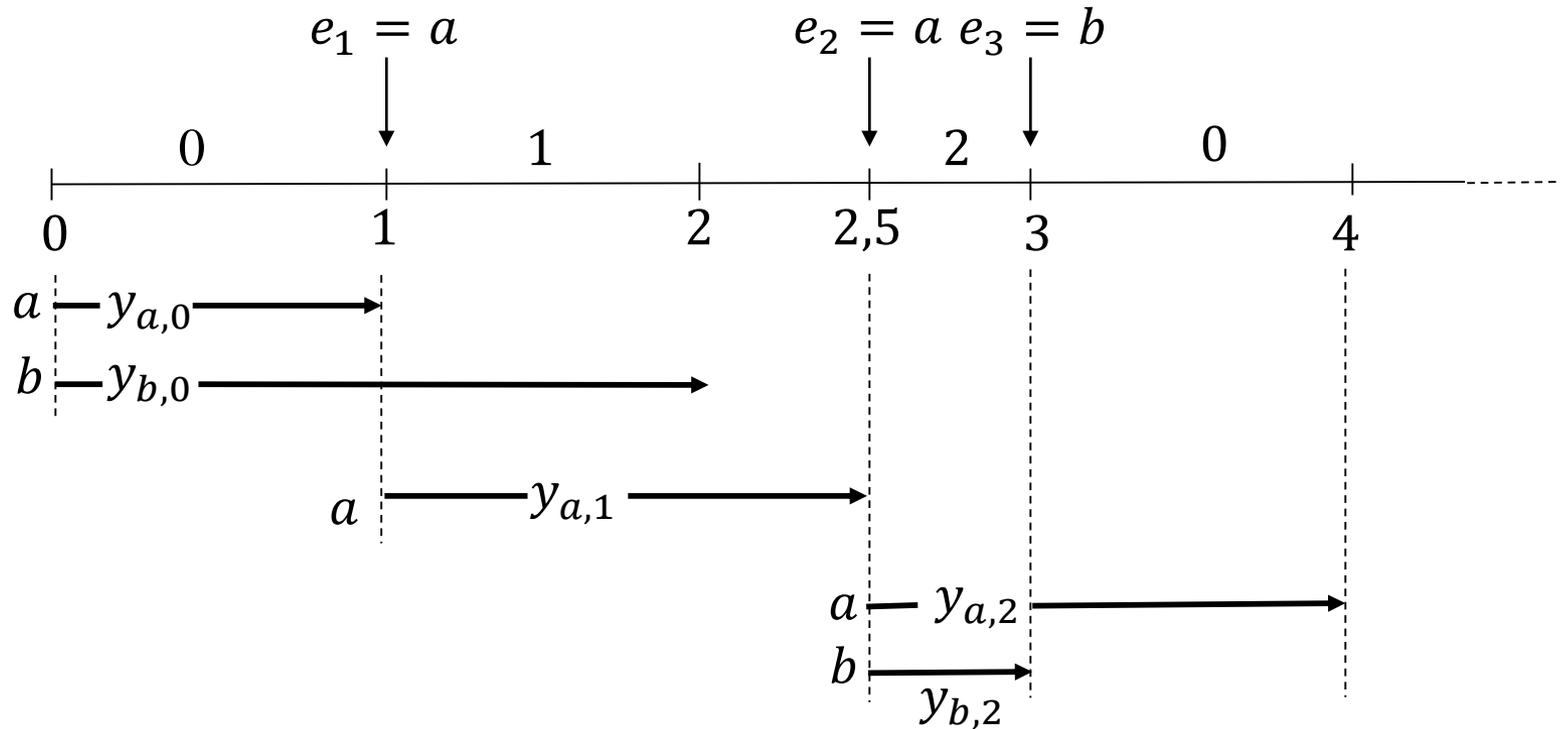
$$y_{a,2} = v_{a,3} = 1,5, \quad y_{b,2} = v_{b,2} = 0,5$$

et on obtient

$$y_2^* = \min \{y_{a,2}, y_{b,2}\} = y_{b,2} \Rightarrow e_3 = b$$

Donc, l'état suivant est

$$x_3 = f^x(2, b) = 0$$



La date de l'événement est  $t_3 = t_2 + y_2^* = 2,5 + y_{b,2} = 3$ .

La dernière transition montrée dans la figure 3.6 survient dans l'état 0. L'événement  $a$  est encore actif, alors que  $b$  était l'événement déclencheur. Seul le compte de  $b$  est incrémenté.

On met à jour les valeurs d'horloges de sorte que

$$y_{a,3} = y_{a,2} - y_2^* = 1, \quad y_{b,3} = v_{b,3} = 1,5$$

et on obtient

$$y_2^* = \min \{ y_{a,3}, y_{b,3} \} = y_{a,3} \Rightarrow e_4 = a$$

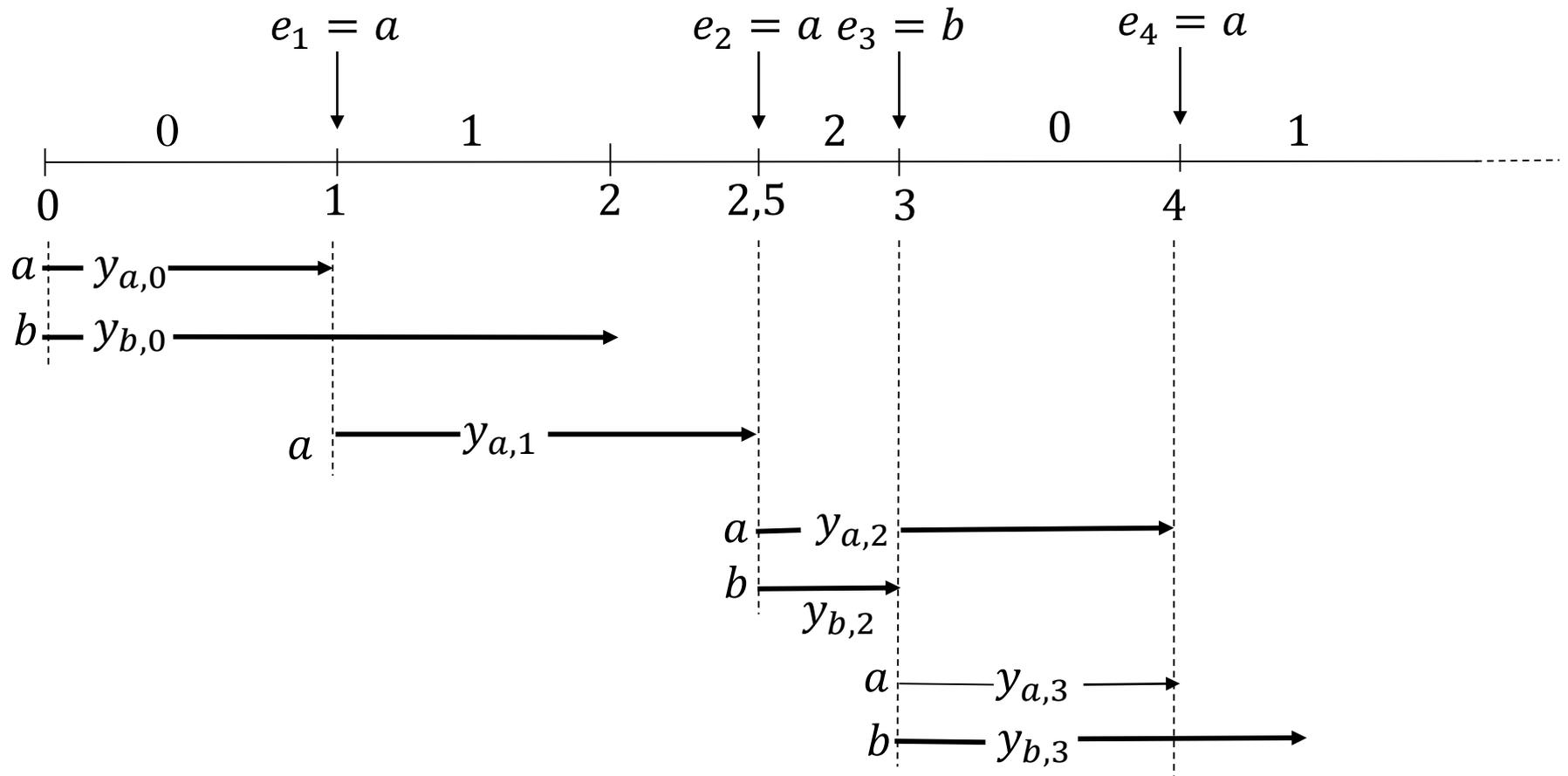


Figure 3.6. Chronogramme pour l'exemple 3.1

L'état suivant est

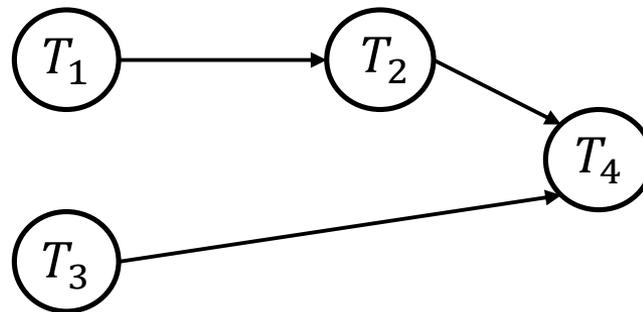
$$x_4 = f^x(0, a) = 1$$

La date de l'événement est  $t_4 = t_3 + y_3^* = 3 + y_{a,3} = 4$ .

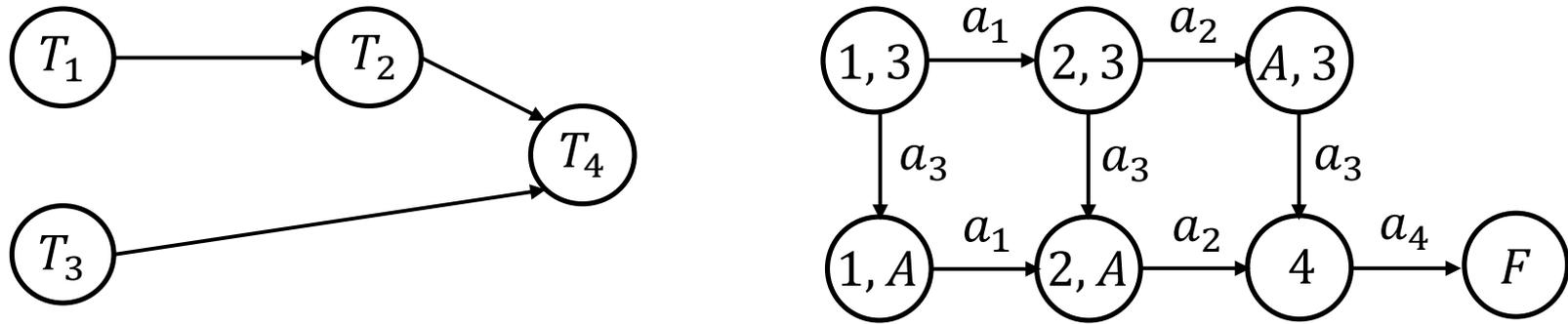
### Exemple 3.1

Beaucoup des processus qui nous intéressent consistent en une variété de tâches à effectuer séquentiellement ou en parallèle ou avec une combinaison de ces deux modes. Chaque tâche réclame une certaine quantité de temps pour être exécutée, ce que nous pouvons modéliser par différentes durées de vies. Il en est ainsi lorsqu'on exécute un algorithme sur un ordinateur ou lors de la manufacture d'un produit.

La figure 3.7, en partie ci-dessous, représente un processus composé de deux tâches en séquence,  $T_1$  et  $T_2$ , qui peuvent être exécutées en parallèle avec une troisième tâche  $T_3$ . On peut supposer que ces tâches sont exécutées sur deux processeurs en parallèle. Les résultats de  $T_2$  et  $T_3$  doivent alors être combinés avant que  $T_4$  ne débute, sur un seul processeur.



On peut décrire ce processus au moyen d'un automate dont le diagramme de transition d'état est aussi montré dans la figure 3.7.

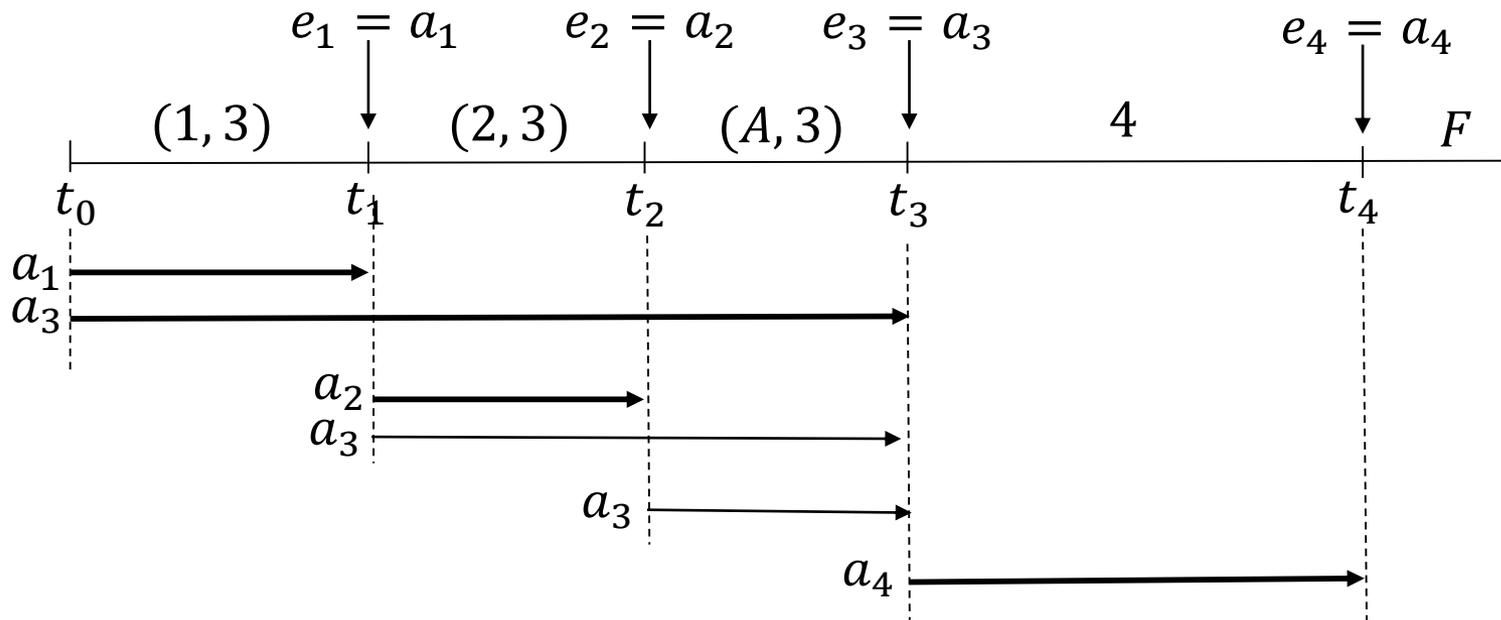


**Figure 3.7.** Graphe de précédence pour l'exemple 3.2 et diagramme de transition d'état

Quatre événements font avancer le processus :  $E = \{a_i, i = 1, \dots, 4\}$ , où  $a_i$  symbolise l'achèvement de la tâche  $i$ . L'espace d'état  $X$  comporte sept états : L'état  $(1, 3)$  indique que  $T_1$  et  $T_3$  sont en train d'être exécutées. Similairement pour l'état  $(2, 3)$ . D'autre part,  $(1, A)$  indique que  $T_1$  est en cours alors que  $T_3$  est « achevée ». Idem avec  $T_2$  pour  $(2, A)$ . L'état  $(A, 3)$  indique que  $T_3$  est en cours tandis que la séquence de tâches  $T_1, T_2$  est achevée. Enfin, l'état  $4$  indique que  $T_4$  est en train d'être exécutée, et l'état  $F$  que le processus est fini. Il est clair que l'événement  $a_i$  est possible seulement lors de l'exécution de la tâche  $i$ .

Dans cet exemple, chaque trajectoire commence à l'état  $(1, 3)$  et se termine avec l'état  $F$ . Chaque événement ne peut arriver qu'une seule fois, aussi la structure d'horloges se réduit à un ensemble de quatre nombres, un pour chaque durée de vie d'événement.

Une trajectoire typique est représentée dans la figure 3.8.



**Figure 3.8.** Un chronogramme typique pour le processus de l'exemple 3.2

L'intervalle de temps  $[t_2, t_3]$  représente un délai de synchronisation pour pouvoir débiter  $T_4$ , qui exige que les résultats de  $T_2$  et  $T_3$  soient disponibles.

Il est important d'observer que l'automate  $(\mathbf{X}, \mathbf{E}, f, \mathbf{\Gamma}, x_0)$  et la structure d'horloges  $\mathbf{V} = \{\mathbf{v}_i: i \in \mathbf{E}\}$  sont distinctement séparées. Cela signifie que  $(\mathbf{X}, \mathbf{E}, f, \mathbf{\Gamma}, x_0)$  est simplement un système ayant besoin d'une séquence d'événements pour l'actionner, et  $\mathbf{V}$  est juste un ensemble de séquences générées indépendamment.

Ainsi,  $\mathbf{V}$  pourrait être utilisée comme entrée pour des automates temporisés différents (dans la mesure où leurs ensembles d'événements sont identiques).

La connexion entre l'automate et sa structure d'horloges se manifeste uniquement au travers de l'ensemble des événements possibles  $\mathbf{\Gamma}(x)$ . Ceci est évident dans les équations d'état (3.11) et (3.12), où  $x$  apparaît seulement à travers  $\mathbf{\Gamma}(x)$ .

Ainsi, le mécanisme de temporisation est capturé par une relation de la forme

$$e' = h(\mathbf{y}, \mathbf{N}, \mathbf{\Gamma}(x))$$

où  $\mathbf{y}$  et  $\mathbf{N}$  peuvent être considérées comme des variables d'état auxiliaires, requises pour mettre à jour l'automate d'état selon

$$x' = f^x(x, e')$$

### 3.2.4. Systèmes de files d'attentes comme automates temporisés

Nous avons déjà développé au chapitre 2 un modèle sous forme d'automate pour une simple station de service de base. Le modèle est le suivant :

$$\begin{aligned} E &= \{a, d\}, \quad X = \{0, 1, 2, \dots\} \\ \Gamma(x) &= \{a, d\} \text{ pour tout } x > 0, \quad \Gamma(0) = \{a\} \\ f(x, e') &= \begin{cases} x + 1 & \text{si } e' = a \\ x - 1 & \text{si } e' = d \text{ et } x > 0 \end{cases} \end{aligned} \quad (3.19)$$

où  $a$  est un événement d'arrivée d'un client, et  $d$  un événement de départ d'un client. L'état  $x$  de l'automate représente la longueur de la file d'attente (incluant un client en train d'être servi). Notons que  $f(x, e')$  n'est pas défini pour  $x = 0$ ,  $e' = d$ , puisque  $d \notin \Gamma(0)$ .

Étant donnée une structure d'horloges  $V$ , l'événement déclencheur est toujours obtenu en comparant les valeurs d'horloges des événements  $a$  et  $d$ , excepté dans le cas où  $x = 0$ , quand seulement un événement  $a$  est possible. Ainsi, on a

$$e' = \begin{cases} d & \text{si } y_d < y_a \text{ et } x > 0 \\ a & \text{sinon} \end{cases}$$

Une fois  $e'$  déterminé, les équations d'état pour les horloges et comptes d'événements (3.11) et (3.12) peuvent être écrites comme suit :

$$y'_a = \begin{cases} y_a - y_d & \text{si } e' = d \\ v_{a, N_a + 1} & \text{sinon} \end{cases} \quad (3.20)$$

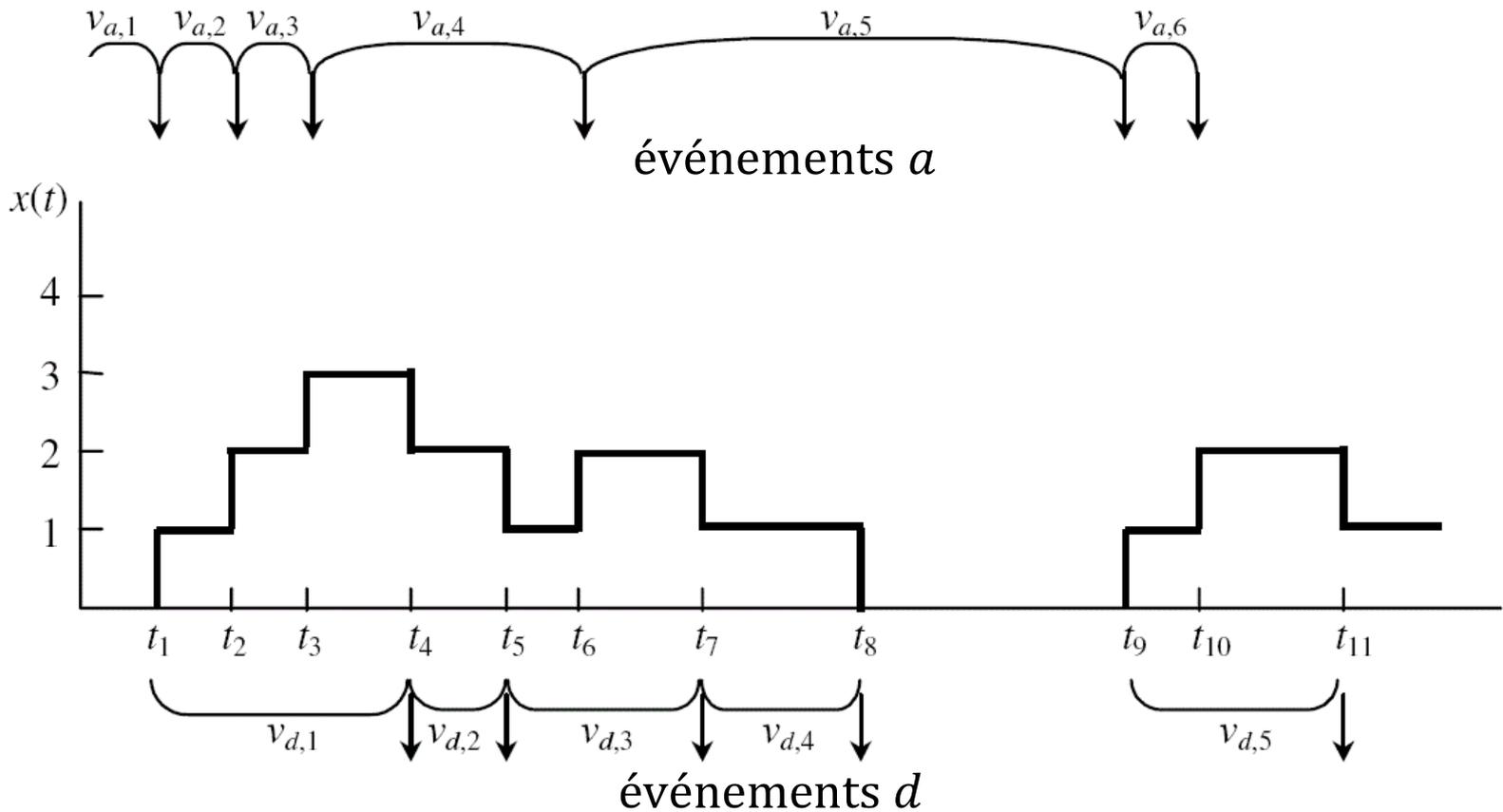
$$y'_d = \begin{cases} y_d - y_a & \text{si } e' = a \text{ et } x > 0 \\ v_{d, N_d + 1} & \text{sinon} \end{cases} \quad (3.21)$$

$$N_a = \begin{cases} N_a + 1 & \text{si } e' = a \\ N_a & \text{sinon} \end{cases} \quad (3.22)$$

$$N_d = \begin{cases} N_d + 1 & \text{si } [e' = a \text{ et } x = 0] \text{ ou } [e' = d \text{ et } x > 1] \\ N_d & \text{sinon} \end{cases} \quad (3.23)$$

À cause de la structure spéciale de ce modèle, il est habituel de représenter les trajectoires en termes d'évolution de la longueur de la file d'attente au fil du temps.

Un exemple est présenté dans la figure 3.9 pour les deux séquences d'horloges de  $a$  et  $d$  montrées dans la figure. Tous les événements  $a$  causent un saut positif d'une unité, et tous les événements  $d$  causent un saut négatif d'une unité.



**Figure 3.9.** Chronogramme d'une station de service de base

La longueur de la file d'attente  $x(t)$  s'accroît de 1 avec chaque arrivée (événement  $a$ ), et décroît de 1 avec chaque départ (événement  $d$ ).

Les activations de  $a$  ont lieu avec chaque occurrence de  $a$ .

Les activations de  $d$  se produisent seulement quand  $d$  survient et la file demeure non-vide, ou quand  $a$  arrive alors que la file est vide.

On observe que le premier événement  $d$  est seulement activé à  $t_1$ . C'est parce qu'initialement  $x = 0$  et  $\Gamma(0) = \{a\}$ . Une observation similaire s'applique au cinquième événement  $d$  à  $t_{11}$ , qui est activé à  $t_9$  suite à la fin de l'intervalle de temps avec file vide  $[t_8, t_9]$ .

À ce point, cela vaut la peine d'insister à nouveau sur les différences entre le modèle animé par les événements – (3.19) à (3.23) – et un modèle animé par le temps pour le même système de file d'attente.

Observons tout d'abord qu'il est difficile d'établir un modèle en temps continu pour ce système basé sur la formulation traditionnelle par équations différentielles. C'est parce que  $x(t)$  est défini sur l'ensemble discret des entiers non-négatifs, et, par conséquent, la dérivée  $\dot{x}(t)$ , à proprement parler, n'existe pas. Nous pouvons cependant établir un modèle à temps discret comme suit.

Soit  $k = 1, 2, \dots$  l'index de temps. La ligne du temps est partitionnée en tranches  $\tau, 2\tau, \dots, k\tau, \dots$  telles qu'au plus une arrivée et un départ peuvent se produire dans une tranche (de longueur  $\tau$ ). Ainsi nous pouvons définir une séquence d'entrée  $a(k)$  telle que, pour  $k = 1, 2, \dots$  :

$$a(k) = \begin{cases} 1 & \text{si une arrivée se produit dans la } k\text{ème tranche de temps} \\ 0 & \text{sinon} \end{cases}$$

Alors, l'équation d'état pour  $x(t)$  est de la forme

$$x(k+1) = \begin{cases} x(k) + 1 & \text{si } a(k) = 1 \text{ et aucun départ ne se produit} \\ & \text{dans la } k\text{ème tranche de temps} \\ x(k) - 1 & \text{si } a(k) = 0 \text{ et un départ se produit} \\ & \text{dans la } k\text{ème tranche de temps} \\ x(k) & \text{sinon} \end{cases}$$

où la condition « départ se produit dans la  $k$ ème tranche de temps » nécessite un analogue de l'état d'horloge  $y_d$  dans (3.21). Établir cet analogue comporte des détails fastidieux que nous omettons, car ce que nous voulons souligner ici est différent.

Spécifiquement, ce modèle animé par le temps impose que  $x(k)$  ne change pas si aucun événement ne se produit dans la  $k$ ème tranche. Si les tranches sont choisies telles que leurs longueurs sont petites, la plupart des tranches ne contiendront probablement pas d'événement. Ce modèle est potentiellement inefficace, puisque **le temps est fréquemment mis à jour alors qu'aucun changement n'a lieu dans l'état du système.**

En outre,  $a(k)$  représente une approximation de la vraie séquence d'horloges des arrivées,  $\mathbf{v}_a$ . Ainsi, on peut vouloir utiliser une très petite valeur de  $t$  afin de satisfaire l'exigence qu'il n'y ait pas deux arrivées dans une même tranche. Mais, alors que cela améliore l'approximation, cela amplifie l'inefficacité mentionnée plus haut.

### 3.2.5. *Le schéma d'ordonnement d'événements*

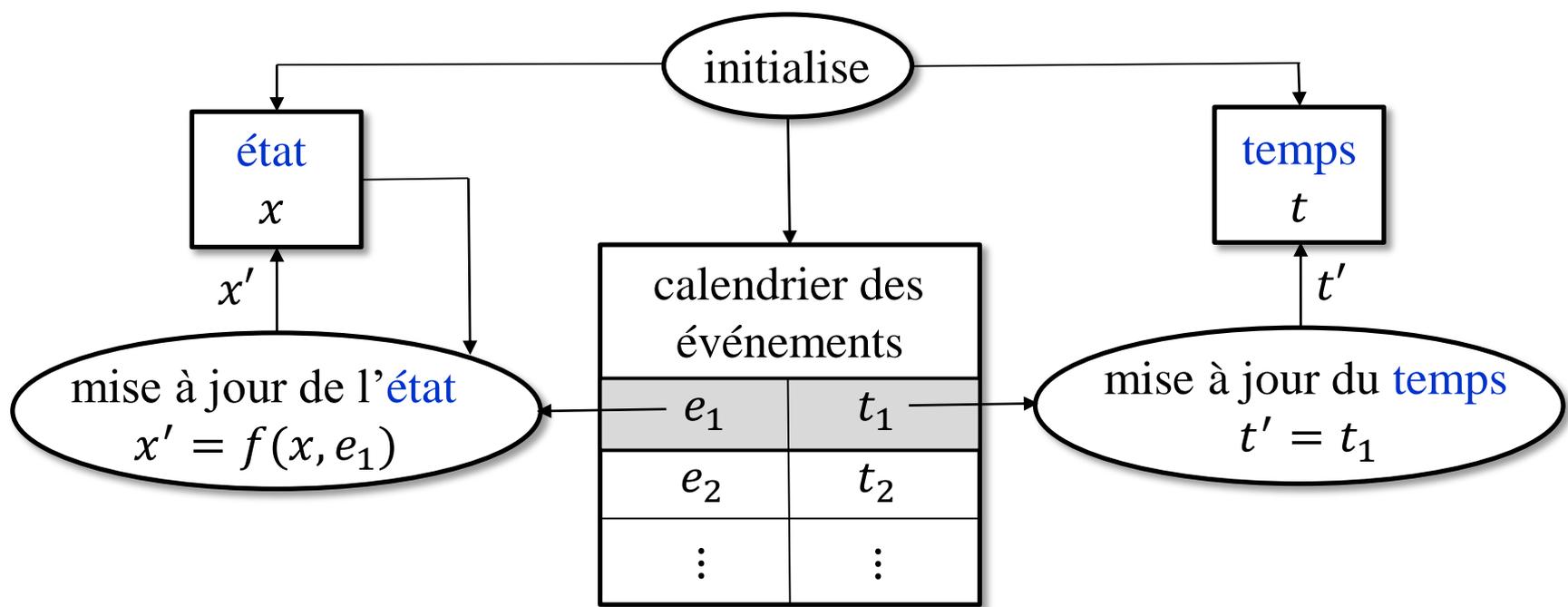
Cette section décrit une variante du modèle d'automate résumé dans la figure 3.4, qui a l'avantage d'être plus simple à programmer. En fait elle constitue la pierre angulaire de la simulation sur ordinateur des systèmes à événements discrets.

Nous faisons la modification suivante. Chaque fois qu'un événement  $i$  est activé à une date  $t_n$  avec un compte  $N_i$ , on planifie que son occurrence suivante est à la date  $(t_n + v_{i,N_i+1})$ . Ainsi, plutôt que de suivre et mettre à jour les valeurs d'horloges  $y_i$ , on maintient une **liste d'événements ordonnancés**

$$L = \{(e_k, t_k)\}, \quad k = 1, 2, \dots, m_L$$

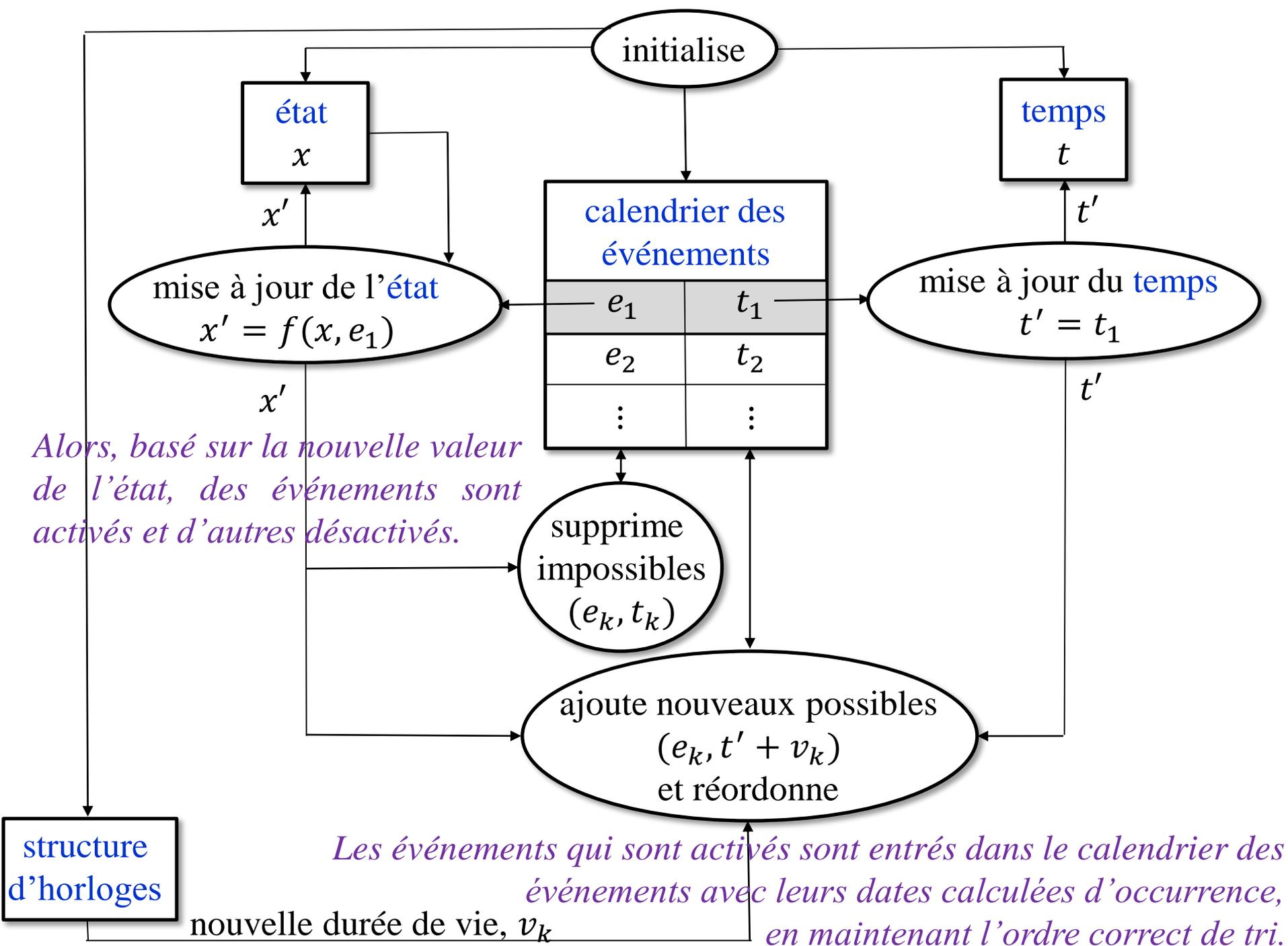
où  $m_L = m$  est le nombre d'événements possibles dans l'état courant. De plus, la liste d'événements ordonnancés est triée dans **l'ordre des dates d'ordonnement croissantes**. Ceci implique que le premier événement dans la liste,  $e_1$ , est toujours l'événement déclencheur.

Ce schéma d'« ordonnancement d'événements » répond simplement à la question du choix d'une mise en forme des données adaptée à la génération du chronogramme d'un SED. Alors que cette variante n'est pas nouvelle conceptuellement par rapport au modèle de la figure 3.4, la **liste d'événements ordonnancés** (aussi appelée le **calendrier des événements**) aide à capturer l'essence du mécanisme de temporisation des événements central à notre discussion.



*Le calendrier des événements contient tous les événements possibles dans l'état courant, triés selon l'ordre des dates d'ordonnement croissantes. L'événement suivant est toujours  $e_1$  et se produit à la date  $t_1$ . Il cause des mises à jour de l'état et du temps. (suite page suivante)*

**Figure 3.10.** Le schéma d'ordonnement d'événements (la figure complète est donnée page suivante)



Le schéma d'ordonnancement des événements doit être compris comme une procédure pour générer des trajectoires dans le temps basées sur le modèle présenté plus haut et prenant en entrée une structure d'horloges. C'est illustré dans la figure 3.10.

L'**état** est initialisé à une valeur donnée  $x_0$ , et le **temps** initial est d'habitude fixé à 0.

La **structure d'horloge** est l'ensemble des séquences d'horloges, une pour chaque événement.

La **liste des événements ordonnancés** est initialisée de sorte à contenir tous les événements possibles dans l'état  $x_0$  avec des dates d'occurrence fournies par les séquences d'horloges appropriées. Elle est aussi triée dans l'ordre croissant des dates.

La procédure est essentiellement identique aux 7 étapes décrites dans la section 3.2.2. Spécifiquement :

**Étape 1.** Enlever le premier élément  $(e_1, t_1)$  de la **liste des événements ordonnancés**.

**Étape 2.** Mettre à jour le **temps** à la nouvelle date d'événement  $t_1$ .

**Étape 3.** Mettre à jour l'état selon la fonction de transition d'état :

$$x' = f(x, e_1).$$

**Étape 4.** Supprimer de la liste des événements ordonnancés les éléments correspondant aux événements impossibles dans le nouvel état, c'est-à-dire supprimer tous les  $(e_k, t_k) \in L$  tels que  $e_k \notin \Gamma(x)$ .

**Étape 5.** Ajouter à la liste des événements ordonnancés tout événement possible qui n'est pas déjà ordonnancé (incluant potentiellement l'événement déclencheur enlevé à l'étape 1). La date calculée d'occurrence pour un tel événement  $i$  est obtenue en additionnant au temps courant une nouvelle durée de vie prise dans la  $i$ ème séquence de la structure d'horloges.

**Étape 6.** Réordonner la liste des événements ordonnancés dans l'ordre des dates d'ordonnancement croissantes.

La procédure est alors répétée à partir de l'étape 1 avec la nouvelle liste ordonnée.

### 3.3 Réseaux de Petri temporisés

Les réseaux de Petri fournissent un cadre général pour construire des modèles non-temporisés de systèmes à événements discrets. Ce cadre peut être étendu aux modèles temporisés en équipant un **réseau de Petri avec une structure d'horloges** et en le transformant ainsi en **réseau de Petri temporisé**.

Les automates temporisés peuvent être obtenus à partir des réseaux de Petri temporisés. Les observations faites au chapitre 2 lors de la discussion des mérites relatifs des réseaux de Petri et des automates continuent de s'appliquer aux versions temporisées de ces deux classes de modèles.

Nous considérons un réseau de Petri marqué  $(P, T, A, w, x)$ , où

- $P$  est un ensemble de places (conditions requises pour autoriser les divers événements),
- $T$  est un ensemble de transitions (événements),
- $A$  contient tous les arcs  $(p_i, t_j)$  tels que  $p_i$  est une place d'entrée de  $t_j$ , et  $(t_j, p_i)$  tels que  $p_i$  est une place de sortie de  $t_j$ ,
- $w(p_i, t_j)$  et  $w(t_j, p_i)$  sont les poids affectés à  $(p_i, t_j)$  et  $(t_j, p_i)$ ,
- $x$  est la fonction de marquage du réseau de Petri, où un marquage représente le nombre de jetons présents dans chaque place du réseau.

Introduisons maintenant une **structure d'horloges** similaire à celle définie pour les automates. La différence est qu'une séquence d'horloges  $\mathbf{v}_j$  est maintenant associée à une **transition**  $t_j$ .

Quand une transition  $t_j$  est permise, il se peut qu'elle ne soit pas immédiatement franchie, mais qu'elle accuse un **retard au franchissement**. Pendant cette durée, les jetons demeurent dans les places d'entrée de  $t_j$ . Le retard au franchissement lorsque la transition  $t_j$  est autorisée pour la  $k$ ème fois, est un nombre réel positif noté  $v_{j,k}$ .

Certaines transitions peuvent ne pas avoir de retard au franchissement : des transitions peuvent être systématiquement franchies aussitôt que permises.

Ainsi  $T$  est partitionné en deux sous-ensembles  $T_0$  et  $T_R$ , tels que  $T = T_0 \cup T_R$ .

- $T_0$  est l'ensemble des transitions sans jamais de retard au franchissement, et
- $T_R$  est l'ensemble des transitions ayant en général un retard au franchissement.

Ces dernières sont appelées **transitions temporisées**.

**Définition :** La **structure d'horloges** associée à un ensemble de transitions temporisées  $T_R \subseteq T$  d'un réseau de Petri marqué  $(P, T, A, w, x)$  est un ensemble  $V = \{\mathbf{v}_j : t_j \in T_R\}$  de séquences d'horloges (ou séquences de durées de vies)

$$\mathbf{v}_j = (v_{j,1}, v_{j,2}, \dots), \quad t_j \in T_R, \quad v_{j,k} \in \mathbb{R}^+, \quad k = 1, 2, \dots$$

Graphiquement, les transitions sans retard au franchissement continuent d'être représentées par des barres, tandis que les transitions temporisées sont représentées par des rectangles. La séquence d'horloges associée à une transition temporisée est normalement indiquée à côté du rectangle.

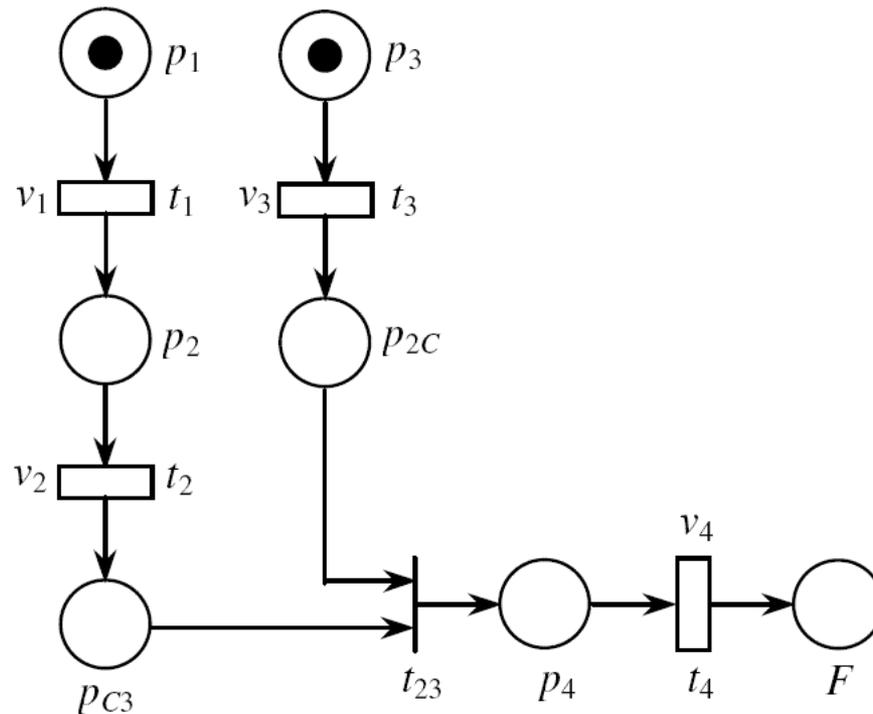
**Définition :** Un réseau de Petri temporisé  $(P, T, A, w, x, V)$  est un sextuplet, où  $(P, T, A, w, x)$  est un réseau de Petri marqué, et  $V = \{v_j: t_j \in T_R\}$  est une structure d'horloges.

**Remarque :** Il faut bien faire la distinction entre **autoriser** une transition et la **franchir**. Une transition temporisée est autorisée dès que ses places d'entrée ont le nombre de jetons nécessaire, mais elle sera franchie après un retard fourni par un élément de sa séquence d'horloges.

### *Exemple 3.3*

Retournons au processus de l'exemple 3.2, où une séquence de deux tâches  $T_1$  et  $T_2$  est exécutée en parallèle avec un troisième tâche  $T_3$ , et ensuite une quatrième tâche  $T_4$  est exécutée qui combine les résultats de  $T_2$  et  $T_3$ . Un modèle de réseau de Petri est montré dans la figure 3.11. Les transitions temporisées  $t_1, \dots, t_4$  correspondent aux événements de fin de tâches et sont représentées par les rectangles accompagnés des retards au franchissement associés  $v_1, \dots, v_4$ .

Quand la tâche 2 est terminée, un jeton est ajouté à la place  $p_{C3}$  (indiquant que  $T_2$  est finie, alors que  $T_3$  peut toujours être en cours). Il en va de même pour la tâche 3 et la place  $p_{2C}$ . Ainsi la transition  $t_{23}$  est autorisée quand les tâches 2 et 3 sont toutes deux achevées (une variante aurait été que les places  $p_{C3}$  et  $p_{2C}$  autorisent directement la transition  $t_4$ , omettant  $t_{23}$  et  $p_4$ ). Le processus est fini quand un jeton est ajouté à la place  $F$ . Dans la figure 3.11 le réseau de Petri est montré dans un état initial tel que les tâches 1 et 3 sont en cours.



**Figure 3.11.** Réseau de Petri temporisé pour l'exemple 3.3

### 3.3.1. Dynamique des réseaux de Petri temporisés

Un modèle de SED basé sur la dynamique temporelle des franchissements de transitions d'un réseau de Petri peut être obtenu exactement comme dans le développement du modèle (3.7) à (3.12), avec les événements remplacés par les transitions.

Cependant, on peut exploiter la structure du réseau de Petri en décomposant le modèle en termes de la **dynamique des transitions individuelles** (événements). Ainsi, les équations d'état dans un tel modèle génèrent des séquences de franchissements de transitions de la forme

$$(\tau_{j,1}, \tau_{j,2}, \dots), \quad j = 1, \dots, m$$

où  $\tau_{j,k}$  est le  $k$ ème instant de franchissement de la transition  $t_j$ ,  $k = 1, 2, \dots$ . Ceci est illustré dans la figure 3.12. Les  $m$  séquences de franchissements de transitions contrastent avec la sortie du modèle sous forme d'automate temporisé de la figure 3.4, constituée d'une unique séquence d'événements temporisés  $((e_1, t_1), (e_2, t_2), \dots)$ .



**Figure 3.12.** Modèle de SED sous forme de réseau de Petri temporisé

Malheureusement, pour les réseaux de Petri arbitraires, établir des équations d'état générales pour tous les instants de franchissement des transitions est vraiment complexe.

Cependant, **pour certaines classes de réseaux de Petri** (par exemple le modèle de la station de service de base, considéré dans la section qui suit), il est assez facile d'obtenir un **modèle pour la dynamique des instants de franchissement des transitions**.

Pour montrer comment de tels modèles peuvent être obtenus, nous nous limitons à une classe restreinte de réseaux de Petri. On considère les réseaux de Petri ayant la propriété que **chaque place a (au plus) une transition en entrée et (au plus) une transition en sortie, et tous les arcs ont un poids égal à 1**.

Un réseau de Petri appartenant à cette classe est appelé un **graphe d'événements**, la terminologie **graphe marqué** est aussi largement utilisée.

Voyons ce qu'implique la définition d'un graphe d'événements.

On exploite d'abord le fait qu'une place  $p_i$  a une seule transition d'entrée  $t_h$ . Soit

$\pi_{i,k}$  : l'instant où la place  $p_i$  reçoit son  $k$ ème jeton,  $k = 1, 2, \dots$

Si  $x(p_i) = 0$  initialement, alors le  $k$ ème instant où un jeton est déposé dans  $p_i$  est précisément le  $k$ ème instant de franchissement de  $t_h$ , que l'on note  $\tau_{h,k}$ .

De façon générale, si  $p_i$  contient initialement  $x_{i0}$  jetons, alors le  $k$ ème instant de franchissement de  $t_h$  est l'instant où  $p_i$  reçoit son  $(x_{i0} + k)$ ème jeton.

Utilisant les notations  $O(t_h)$ , pour l'ensemble des places de sortie d'une transition  $t_h$ , et  $x_{i0}$  pour le marquage initial de la place  $p_i$ , on a donc

$$\pi_{i,k+x_{i0}} = \tau_{h,k}, \quad p_i \in O(t_h), \quad k = 1, 2, \dots \quad (3.24)$$

et

$$\pi_{i,k} = 0 \quad \text{pour tout} \quad k = 1, \dots, x_{i0} \quad (3.25)$$

On note que (3.24) s'écrit de manière équivalente

$$\pi_{i,k} = \tau_{h,k-x_{i0}}, \quad p_i \in O(t_h), \quad k = x_{i0} + 1, x_{i0} + 2, \quad (3.26)$$

On exploite maintenant le fait qu'une place a au plus une transition de sortie.

On considère d'abord le cas d'une transition  $t_j$  ayant une seule place d'entrée,  $p_i$ .

- si  $t_j$  n'est pas temporisée ( $t_j \in \mathbf{T}_0$ ), alors le  $k$ ème instant de franchissement de  $t_j$ , noté  $\tau_{j,k}$ , est précisément l'instant où  $p_i$  reçoit son  $k$ ème jeton, et donc autorise  $t_j$ .

On a donc la relation toute simple :

$$\tau_{j,k} = \pi_{i,k}, \quad k = 1, 2, \dots$$

- si  $t_j$  est temporisée ( $t_j \in \mathbf{T}_R$ ), avec la séquence d'horloges  $\mathbf{v}_j$ , alors cette relation devient

$$\tau_{j,k} = \pi_{i,k} + v_{i,k}, \quad k = 1, 2, \dots$$

On considère maintenant le cas d'une transition  $t_j$  avec plusieurs places d'entrée,  $p_i \in I(t_j)$ , où  $I(t_j)$  est l'ensemble des places d'entrée de la transition  $t_j$ .

La transition  $t_j$  est autorisée pour la  $k$ ème fois au moment de la dernière acquisition par une place de  $I(t_j)$  de son  $k$ ème jeton. Notant  $p_l \in I(t_j)$  cette place, l'autorisation de  $t_j$  se produit ainsi à l'instant  $\pi_{l,k}$  tel que

$$\pi_{l,k} \geq \pi_{i,k} \quad \text{pour tout } p_i \in I(t_j).$$

On a donc dans les deux cas

- si  $t_j$  n'est pas temporisée ( $t_j \in \mathbf{T}_0$ ),

$$\tau_{j,k} = \max_{p_i \in I(t_j)} \{\pi_{i,k}\} \quad (3.27)$$

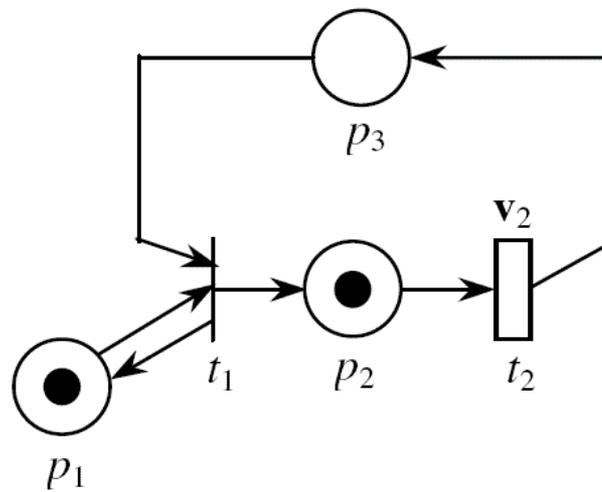
- si  $t_j$  est temporisée ( $t_j \in \mathbf{T}_R$ ),

$$\tau_{j,k} = \max_{p_i \in I(t_j)} \{\pi_{i,k}\} + v_{j,k}, \quad k = 1, 2, \dots \quad (3.28)$$

En résumé, la combinaison de (3.24) à (3.28) fournit un système d'équations récurrentes permettant de déterminer les dates de franchissement des transitions pour les graphes d'événements, une classe particulière de réseaux de Petri.

### *Exemple 3.4*

On considère le réseau de Petri de la figure 3.13. C'est un graphe d'événements puisque chaque place a un arc d'entrée et un arc de sortie de poids 1.



**Figure 3.13.** Réseau de Petri temporisé pour l'exemple 3.4

Un système d'équations récurrentes de la forme (3.24) à (3.28) peut donc être obtenu. Par inspection, on peut immédiatement écrire les équations suivantes pour  $\tau_{1,k}$  et  $\tau_{2,k}$  :

$$\tau_{1,k} = \max\{\pi_{1,k}, \pi_{3,k}\} \quad k = 1, 2, \dots \quad (3.29)$$

$$\tau_{2,k} = \pi_{2,k} + v_{2,k} \quad k = 1, 2, \dots \quad (3.30)$$

puisque, d'une part, la transition  $t_1$  n'est pas temporisée et est franchie pour la  $k$ ème fois quand la dernière de ses places d'entrée acquiert un  $k$ ème jeton et, d'autre part, la transition  $t_2$  est temporisée et est franchie après un retard  $v_{2,k}$  suivant la  $k$ ème acquisition d'un jeton par  $p_2$ .

L'état initial est  $[1, 1, 0]$ . Par inspection, on peut donc aussi écrire :

$$\pi_{1,k} = \tau_{1,k-1} \quad k = 2, 3, \dots, \quad \pi_{1,1} = 0 \quad (3.31)$$

puisque la place  $p_1$  contient un jeton initialement et doit atteindre le prochain franchissement de la transition  $t_1$  pour le réacquérir.

De même, pour  $p_2$ ,

$$\pi_{2,k} = \tau_{1,k-1} \quad k = 2, 3, \dots, \quad \pi_{2,1} = 0 \quad (3.32)$$

Enfin,  $p_3$  reçoit un jeton chaque fois que  $t_2$  est franchie, et puisqu'elle a commencé sans jeton, on a

$$\pi_{3,k} = \tau_{2,k} \quad k = 1, 2, \dots \quad (3.33)$$

Combinant (3.29) à (3.33), on peut éliminer  $\pi_{1,k}$ ,  $\pi_{2,k}$  et  $\pi_{3,k}$  et obtenir les équations récurrentes recherchées pour  $\tau_{1,k}$  et  $\tau_{2,k}$ .

On obtient, pour  $k = 1, 2, \dots$ ,

$$\begin{aligned} \tau_{1,k} &= \max\{\tau_{1,k-1}, \tau_{1,k-1} + v_{2,k}\} \\ &= \tau_{1,k-1} + v_{2,k}, \quad \tau_{1,0} = 0 \end{aligned} \quad (3.34)$$

$$\tau_{2,k} = \tau_{1,k-1} + v_{2,k} \quad k = 1, 2, \dots \quad (3.35)$$

Ces équations récurrentes correspondent au fait, somme toute évident, que les deux transitions attendent pour le retard au franchissement au niveau de  $t_2$ .

Pour des réseaux de Petri généraux, il demeure possible d'obtenir des équations de la forme (3.24) à (3.28) pour les dates de franchissement des transitions et les dates d'autorisation des places.

Cependant, les relations récurrentes deviennent nettement plus complexes.

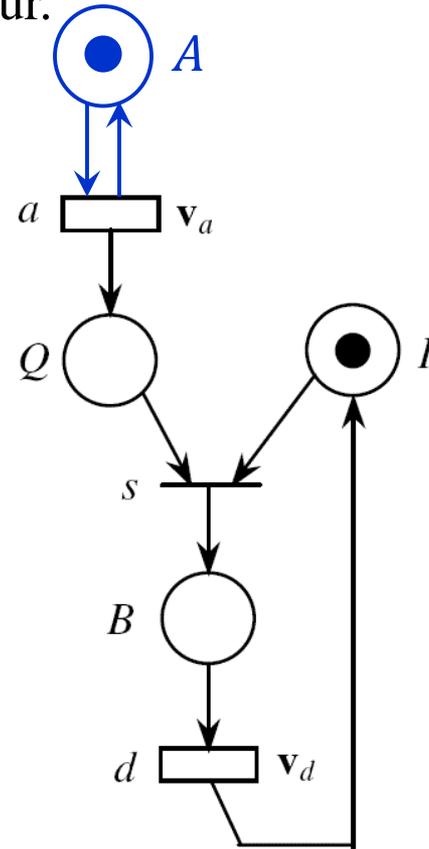
### 3.3.2. Systèmes de files d'attentes vus comme réseaux de Petri temporisés

On se rappelle le modèle sous forme de réseau de Petri d'une station de service de base présenté au chapitre 2, avec un ensemble de places  $\mathbf{P} = \{Q, I, B\}$ .

Une version temporisée de ce modèle est montrée dans la figure 3.14 dans l'état  $x = [0, 1, 0]$ , c'est-à-dire quand la file est vide et le serveur est inactif.

Dans ce cas, l'ensemble des transitions temporisées est  $\mathbf{T}_R = \{a, d\}$ , correspondant aux arrivées des clients et aux départs du serveur.

La transition  $s$ , par contre, n'est pas sujette au retard au franchissement : un service démarre dès que le serveur est inactif et qu'un client est dans la file.



**Figure 3.14.** Modèle réseau de Petri temporisé pour la station de service de base.

La structure d'horloges pour ce modèle est composée des séquences  $\mathbf{v}_a = (v_{a,1}, v_{a,2}, \dots)$  et  $\mathbf{v}_d = (v_{d,1}, v_{d,2}, \dots)$ . C'est la même structure d'horloges que pour le modèle sous forme d'automate temporisé présenté précédemment.

En outre, notant  $x(t) = x(Q) + x(B)$  le nombre total de jetons dans les places  $Q$  et  $B$  à l'instant  $t$ , la figure 3.9 représente une trajectoire typique de cette variable dans le cadre des réseaux de Petri temporisés.

Considérant la figure 3.14, on voit en considérant le nombre d'arcs d'entrée et de sortie des places que le modèle est un graphe d'événements. Cela permet d'obtenir facilement des équations de la forme (3.24) à (3.28) pour décrire la dynamique temporelle de franchissement des transitions de la station de service de base.

Les variables temporelles associées aux transitions sont

$a_k$  :  $k$ ème date d'arrivée

$d_k$  :  $k$ ème date de départ

$s_k$  :  $k$ ème date de début de service

et celles associées aux places

$\pi_{Q,k}$  : instant où  $Q$  reçoit son  $k$ ème jeton

$\pi_{I,k}$  : instant où  $I$  reçoit son  $k$ ème jeton

$\pi_{B,k}$  : instant où  $B$  reçoit son  $k$ ème jeton

avec la condition initiale  $\pi_{I,1} = 0$ .

**Remarque** : Dans un réseau de Petri non-temporisé, lorsqu'une transition est autorisée, elle n'est pas nécessairement immédiatement franchie; lorsque plusieurs transitions sont autorisées, tous les cas de figure sont possibles en ce qui concerne l'ordre de franchissement des transitions. L'analyse qualitative d'un réseau de Petri envisage tous les ordres possibles.

En contraste, dans un réseau de Petri temporisé, lorsqu'une transition est autorisée, elle est soit immédiatement franchie, si elle est non-temporisée, soit franchie avec un retard au franchissement défini par la valeur appropriée dans la séquence d'horloges, si elle est temporisée.

Si la **place A** n'était pas introduite dans la figure 3.14, les horloges (ou minuterics) associées à la transition  $a$  seraient toutes initialisées à l'instant 0 et lancées en parallèle (pas de condition d'autorisation restreignant les franchissements). Les événements correspondant aux horloges de la séquence d'horloges seraient actifs simultanément. Initialement on aurait autant d'événements actifs de type  $a$  que d'éléments dans la séquence  $\mathbf{v}_a = (v_{a,1}, v_{a,2}, \dots)$ . Il faudrait généraliser l'ensemble des événements actifs à la notion de multi-ensemble, permettant de prendre en compte la multiplicité d'événements de même type. La séquence d'horloges  $\mathbf{v}_a$  serait, aussi, la séquence des dates d'occurrence des événements de type  $a$ ,  $(\tau_{a,1}, \tau_{a,2}, \dots)$ .

La **place A** dans la figure 3.14 permet de « séquentialiser » les initialisations et lancements des horloges (ou minuterics) associées à la transition  $a$ .

Pour la transition  $a$  et la place  $A$ , on a

$$\begin{aligned} a_k &= \pi_{A,k} + v_{a,k}, & k &= 1, 2, \dots \\ \pi_{A,k} &= a_{k-1}, & k &= 2, 3, \dots, & \pi_{A,1} &= 0 \end{aligned}$$

On a donc pour la transition  $a$

$$a_k = a_{k-1} + v_{a,k}, \quad k = 1, 2, \dots, \quad a_0 = 0 \quad (3.36)$$

et pour les transitions  $s$  et  $d$ , appliquant (3.27) et (3.28),

$$s_k = \max\{\pi_{Q,k}, \pi_{I,k}\}, \quad k = 1, 2, \dots \quad (3.37)$$

$$d_k = \pi_{B,k} + v_{d,k}, \quad k = 1, 2, \dots \quad (3.38)$$

On a aussi pour les places, appliquant (3.25) et (3.26),

$$\pi_{Q,k} = a_k, \quad k = 1, 2, \dots \quad (3.39)$$

$$\pi_{I,k} = d_{k-1}, \quad k = 2, 3, \dots, \quad \pi_{I,1} = 0 \quad (3.40)$$

$$\pi_{B,k} = s_k, \quad k = 1, 2, \dots \quad (3.41)$$

Combinant (3.37) à (3.41) pour éliminer  $\pi_{Q,k}$ ,  $\pi_{I,k}$  et  $\pi_{B,k}$ , on obtient

$$s_k = \max\{a_k, d_{k-1}\}, \quad k = 1, 2, \dots, \quad d_0 = 0 \quad (3.42)$$

$$d_k = s_k + v_{d,k}, \quad k = 1, 2, \dots \quad (3.43)$$

On peut aller plus loin et éliminer  $s_k$  de (3.43) pour obtenir l'importante relation, fondamentale pour ce système élémentaire de file d'attente :

$$d_k = \max\{a_k, d_{k-1}\} + v_{d,k}, \quad k = 1, 2, \dots, \quad d_0 = 0 \quad (3.44)$$

Cette relation récurrente simple caractérise les dates de départ des clients.

Elle capture le fait que le  $k$ ème départ se produit  $v_{d,k}$  unités de temps après le  $(k - 1)$ ème départ, excepté quand  $a_k > d_{k-1}$ . Ce dernier cas se produit quand le départ à  $d_{k-1}$  laisse la file vide; le serveur doit alors attendre la prochaine arrivée à l'instant  $a_k$ , et générer le prochain départ à l'instant  $a_k + v_{d,k}$ .

Réécrivant (3.36) et (3.44) pour  $k = 1, 2, \dots$  comme suit

$$a_k = a_{k-1} + v_{a,k}, \quad a_0 = 0 \quad (3.45)$$

$$d_k = \max\{a_k, d_{k-1}\} + v_{d,k}, \quad d_0 = 0 \quad (3.46)$$

On obtient un modèle d'espace d'état dans le cadre de la figure 3.12. C'est un système d'équations récurrentes pour les instants de franchissements de transitions déduit du modèle sous forme de réseau de Petri du système de file d'attente.

Le modèle du système de file d'attente est activé par les séquences d'horloges  $\mathbf{v}_a$  et  $\mathbf{v}_d$ , et sa sortie est composée des séquences des dates d'arrivée et de départ  $(a_1, a_2, \dots)$  et  $(d_1, d_2, \dots)$  générées au moyen des équations d'état (3.36) et (3.44).

### 3.3.3. Structure des équations récurrentes pour les graphes d'événements

On suppose pour le moment qu'il y a initialement au maximum un jeton par place; cette hypothèse sera relaxée plus tard.

Les variables apparaissant dans le système d'équations récurrentes peuvent être partitionnées en trois groupes : les dates associées aux places à jeton, celles associées aux places sans jeton, et celles associées aux transitions.

Pour l'exemple de la station de service de base, cela donne :

$$\begin{array}{lll} \pi_{A,k+1} = a_k, & k = 1, 2, \dots, & \pi_{A,1} = 0 \\ \pi_{I,k+1} = d_k, & k = 1, 2, \dots, & \pi_{I,1} = 0 \end{array}$$

$$\begin{array}{ll} \pi_{Q,k} = a_k, & k = 1, 2, \dots \\ \pi_{B,k} = s_k, & k = 1, 2, \dots \end{array}$$

$$\begin{array}{ll} a_k = \pi_{A,k} + v_{a,k}, & k = 1, 2, \dots \\ s_k = \max\{\pi_{Q,k}, \pi_{I,k}\}, & k = 1, 2, \dots \\ d_k = \pi_{B,k} + v_{d,k}, & k = 1, 2, \dots \end{array}$$

Les variables d'état sont les dates associées aux places à jeton. Pour résoudre le système d'équations récurrentes, on réécrit les équations en éliminant les autres variables. Ces variables, en particulier les dates de franchissement des transitions, peuvent être vues comme des variables auxiliaires, qui seront facilement obtenues à partir des variables d'état une fois que celles-ci seront calculées.

Pour éliminer les variables auxiliaires, il est en général plus pratique de remplacer la **fonction max** par une **opération**. On utilise le formalisme (d'algèbre de chemins) déjà rencontré pour les expressions régulières.

Le **max** de deux variables correspond à l'opération binaire  $\vee$ , qui est commutative ( $a \vee b = b \vee a$ ), associative ( $a \vee (b \vee c) = (a \vee b) \vee c$ ), idempotente ( $a \vee a = a$ ), et possède un élément neutre, qui est  $-\infty$  pour le max, ( $a \vee -\infty = -\infty \vee a = a$ ).

L'addition de deux variables correspond à l'opération binaire  $.$ , qui est associative ( $a.(b.c) = (a.b).c$ ), possède un élément neutre, qui est ici 0, ( $a.0 = 0.a = a$ ), et est distributive par rapport à  $\vee$  ( $a.(b \vee c) = a.b \vee a.c$  et  $(b \vee c).a = b.a \vee c.a$ ).

La fermeture transitive n'est pas nécessaire ici, car on n'aura pas de dépendances circulaires entre variables auxiliaires (de même indice d'itération,  $k$ ) ou entre variables d'état (de même indice,  $k + 1$ ). (Note : la fermeture transitive serait  $a^* = a$ , si  $a \leq 0$ , et  $+\infty$ , si  $a > 0$ ).

Notons, pour les calculs,  $a$  pour  $a_k$ ,  $\pi_Q$  pour  $\pi_{Q,k}$ ,  $\pi'_A$  pour  $\pi_{A,k+1}$ , etc.

On a

$$\pi'_A = a, \quad \pi'_I = d,$$

$$\pi_Q = a, \quad \pi_B = s,$$

$$a = \pi_A \cdot v_a, \quad s = \pi_Q \vee \pi_I, \quad d = \pi_B \cdot v_d$$

On substitue les expressions pour  $a$  et  $d$  dans les deux premières équations :

$$\pi'_A = \pi_A \cdot v_a, \quad \pi'_I = \pi_B \cdot v_d.$$

On substitue l'expression pour  $\pi_B$  dans la seconde équation :

$$\pi'_I = s \cdot v_d,$$

puis l'expression pour  $s$ ,

$$\pi'_I = (\pi_Q \vee \pi_I) \cdot v_d,$$

et enfin l'expression pour  $\pi_Q$  en fonction de  $\pi_A$ ,

$$\pi'_I = (\pi_A \cdot v_a \vee \pi_I) \cdot v_d.$$

On obtient donc

$$\pi'_A = \pi_A \cdot v_a$$

$$\pi'_I = \pi_A \cdot v_a \cdot v_d \vee \pi_I \cdot v_d$$

soit encore, sous forme matricielle,

$$\begin{bmatrix} \pi'_A & \pi'_I \end{bmatrix} = \begin{bmatrix} \pi_A & \pi_I \end{bmatrix} \begin{bmatrix} v_a & v_a \cdot v_d \\ -\infty & v_d \end{bmatrix}$$

Une fois résolues les équations récurrentes pour l'état,

$$\pi_{A,k+1} = \pi_{A,k} + v_{a,k}, \quad k = 1, 2, \dots, \quad \pi_{A,1} = 0,$$

$$\pi_{I,k+1} = \max\{\pi_{A,k} + v_{a,k}, \pi_{I,k}\} + v_{d,k}, \quad k = 1, 2, \dots, \quad \pi_{I,1} = 0,$$

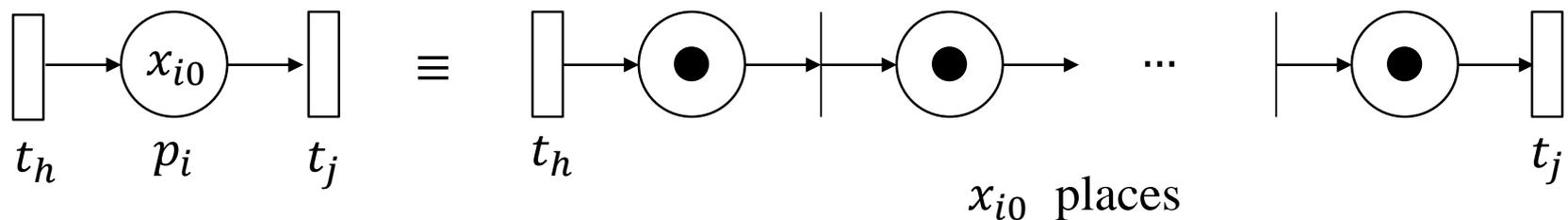
on obtient facilement les séquences des dates de franchissement des transitions

$$a_k = \pi_{A,k} + v_{a,k}, \quad k = 1, 2, \dots$$

$$s_k = \max\{\pi_{A,k} + v_{a,k}, \pi_{I,k}\}, \quad k = 1, 2, \dots$$

$$d_k = \max\{\pi_{A,k} + v_{a,k}, \pi_{I,k}\} + v_{d,k}, \quad k = 1, 2, \dots$$

**Remarque :** Dans le cas où certaines places contiennent initialement plus d'un jeton, elles peuvent être remplacées par autant de places, contenant chacune un jeton, qu'il y a de jetons, ces places étant interconnectées par des transitions non-temporisées :



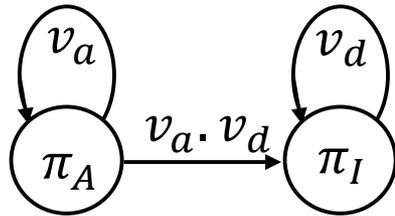
On se ramène ainsi au cas précédent, avec un vecteur d'état de taille  $\sum_{p_i \in P} x_{i0}$ , et, en résolvant les équations récurrentes pour cet état, on en déduit les séquences des dates de franchissement des transitions.

### 3.3.4. Graphes d'événements avec retards au franchissement constants

Pour introduire la méthode, reprenons l'exemple de la station de service de base.  
 Au système d'équations récurrentes pour les places marquées

$$[\pi'_A \quad \pi'_I] = [\pi_A \quad \pi_I] \begin{bmatrix} v_a & v_a \cdot v_d \\ -\infty & v_d \end{bmatrix}$$

est associé le graphe



Lorsque les retards au franchissement sont constants, indépendants de  $k$ , on a donc

$$\begin{aligned}
 [\pi_{A,k+1} \quad \pi_{I,k+1}] &= [\pi_{A,1} \quad \pi_{I,1}] \cdot \begin{bmatrix} v_a & v_a \cdot v_d \\ -\infty & v_d \end{bmatrix}^k \\
 &= [0 \quad 0] \cdot \begin{bmatrix} v_a & v_a \cdot v_d \\ -\infty & v_d \end{bmatrix}^k \quad \rightarrow \text{chemins de longueur } k \\
 &= [0 \quad 0] \cdot \begin{bmatrix} v_a^k & v_a^k \cdot v_d \vee v_a^{k-1} \cdot v_d^2 \vee \dots \vee v_a \cdot v_d^k \\ -\infty & v_d^k \end{bmatrix} \\
 &= [v_a^k \vee -\infty \quad (v_a^k \cdot v_d \vee v_a \cdot v_d^k) \vee v_d^k] \\
 &= [v_a^k \quad v_a^k \cdot v_d \vee v_a \cdot v_d^k]
 \end{aligned}$$

Ainsi

$$\begin{aligned} [\pi_{A,k+1} \quad \pi_{I,k+1}] &= [v_a^k \quad v_a^k \cdot v_d] \quad \text{si } v_a \geq v_d \\ &= [v_a^k \quad v_a \cdot v_d^k] \quad \text{si } v_a \leq v_d \end{aligned}$$

Abandonnant provisoirement la notation « chemin »,  $(V, \cdot)$ , et revenant au formalisme  $(\max, +)$ , on a donc, si  $v_a \geq v_d$ ,

$$\pi_{A,k} = (k - 1)v_a, \quad k \geq 1$$

$$\pi_{I,k} = (k - 1)v_a + v_d, \quad k \geq 1$$

$$a_k = kv_a, \quad k \geq 1$$

$$s_k = kv_a, \quad k \geq 1$$

$$d_k = kv_a + v_d, \quad k \geq 1,$$

et, si  $v_a \leq v_d$ ,

$$\pi_{A,k} = (k - 1)v_a, \quad k \geq 1$$

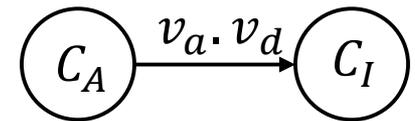
$$\pi_{I,k} = v_a + (k - 1)v_d, \quad k \geq 1$$

$$a_k = kv_a, \quad k \geq 1$$

$$s_k = v_a + (k - 1)v_d, \quad k \geq 1$$

$$d_k = v_a + kv_d, \quad k \geq 1.$$

Le graphe du système d'équations récurrentes a la structure :



où la composante  $C_A$  précède la composante  $C_I$ . Alors que

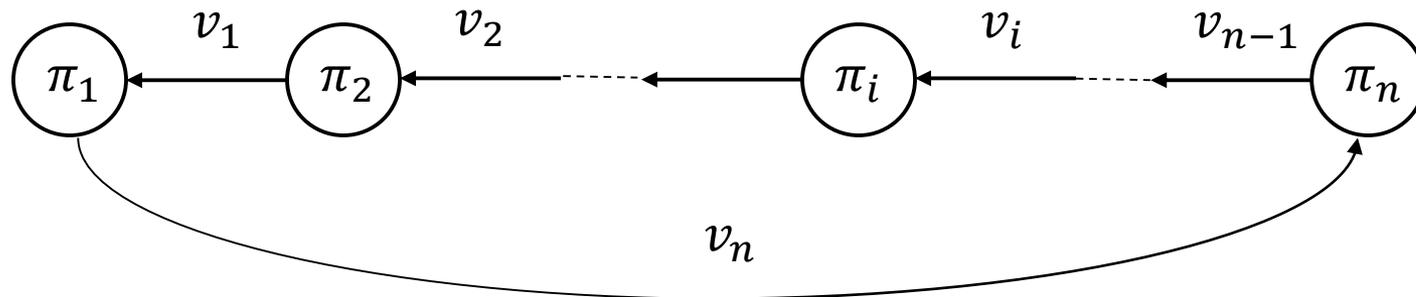
$C_A$  suit son propre rythme, celui de  $C_I$  est imposé par celui de  $C_A$  si  $v_a \geq v_d$ .

Les **composantes fortement connexes** d'un graphe orienté sont composées par les sous-ensembles de nœuds tels qu'il existe un chemin de chaque nœud vers chaque autre nœud du sous-ensemble.

De façon générale, si, dans le graphe du système d'équations récurrentes pour les dates associées aux places marquées, on remplace chaque composante fortement connexe par un nœud la représentant, on obtient un **graphe acyclique** dont la structure met en évidence les **précédences entre composantes fortement connexes**. Cette structure est exploitée pour résoudre le système d'équations récurrentes.

Considérons une composante fortement connexe.

Supposons tout d'abord qu'elle est constituée d'**un circuit** :



Le système d'équations récurrentes est, avec la notation simplifiée,

$$[\pi'_1 \quad \pi'_2 \quad \dots \quad \pi'_i \quad \dots \quad \pi'_n] = [\pi_1 \quad \pi_2 \quad \dots \quad \pi_i \quad \dots \quad \pi_n] A$$

où la matrice  $A$  du système a une structure particulière.

$$A = \begin{bmatrix} -\infty & -\infty & \dots & -\infty & \dots & -\infty & v_n \\ v_1 & -\infty & & -\infty & & -\infty & -\infty \\ -\infty & v_2 & & -\infty & & -\infty & -\infty \\ \vdots & & & & & & \vdots \\ -\infty & -\infty & & v_i & & -\infty & -\infty \\ \vdots & & & & & & \vdots \\ -\infty & -\infty & & -\infty & & v_{n-1} & -\infty \end{bmatrix}$$

Cette matrice a la propriété suivante :

$$A^n = \begin{bmatrix} v_1 \cdot v_2 \dots v_n & -\infty & \dots & -\infty \\ -\infty & v_2 \dots v_n \cdot v_1 & \dots & -\infty \\ \vdots & \vdots & \dots & \vdots \\ -\infty & -\infty & \dots & v_n \cdot v_1 \dots v_{n-1} \end{bmatrix}$$

soit, comme la multiplication ( $\leftrightarrow +$ ) est commutative,

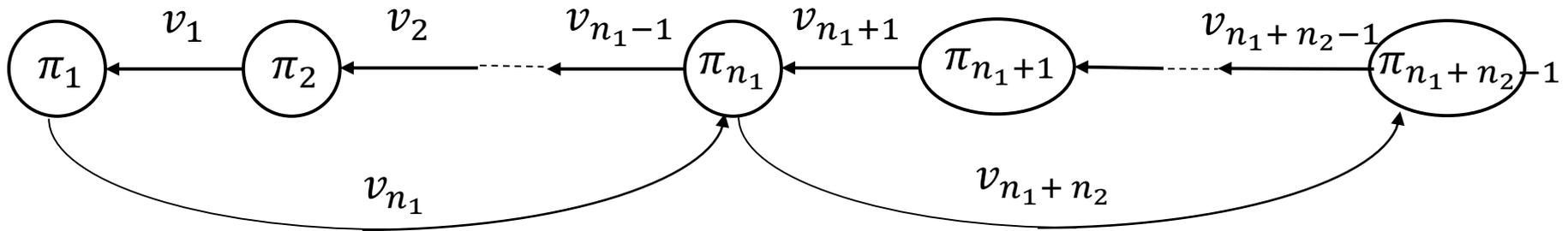
$$A^n = (v_1 \cdot v_2 \dots v_n) \cdot \begin{bmatrix} 0 & -\infty & \dots & -\infty \\ -\infty & 0 & \dots & -\infty \\ \vdots & \vdots & \dots & \vdots \\ -\infty & -\infty & \dots & 0 \end{bmatrix}$$

d'où l'on déduit

$$[\pi_{1,k+n} \ \pi_{2,k+n} \ \dots \ \pi_{i,k+n} \ \dots \ \pi_{n,k+n}] = (v_1 \cdot v_2 \dots v_n) \cdot [\pi_{1,k} \ \pi_{2,k} \ \dots \ \pi_{i,k} \ \dots \ \pi_{n,k}]$$

Les dates  $\pi_{i,k}$ ,  $i = 1, \dots, n$ , satisfont toutes  $\pi_{i,k+n} = \pi_{i,k} + u$ , avec  $u = \sum_{i=1}^n v_i$ . Les séquences de dates sont périodiques, de même période  $n$ . L'incrément associé à une période est identique pour toutes les dates et est égal à la somme des retards le long du circuit,  $\sum_{i=1}^n v_i$ .

Considérons maintenant une composante fortement connexe composée de **deux circuits** selon



La matrice du système est

$$A = \begin{bmatrix} -\infty & \dots & -\infty & v_{n_1} & -\infty & \dots & -\infty \\ v_1 & \dots & -\infty & -\infty & -\infty & \dots & -\infty \\ \vdots & \dots & \dots & \vdots & \vdots & \dots & -\infty \\ -\infty & \dots & v_{n_1-1} & -\infty & -\infty & \dots & v_{n_1+n_2} \\ -\infty & \dots & -\infty & v_{n_1+1} & -\infty & -\infty & -\infty \\ \vdots & \dots & \dots & \vdots & \vdots & \dots & \vdots \\ -\infty & \dots & -\infty & -\infty & \dots & v_{n_1+n_2-1} & -\infty \end{bmatrix}$$

Il existe  $k_0$  et  $n$ , entiers positifs, et une durée  $u$  tels que

$$[\pi_{1,k_0} \ \pi_{2,k_0} \ \dots \ \pi_{n_1+n_2-1,k_0}] A^n = u \cdot [\pi_{1,k_0} \ \pi_{2,k_0} \ \dots \ \pi_{n_1+n_2-1,k_0}].$$

Le vecteur

$$[\pi_{1,k_0} \ \pi_{2,k_0} \ \dots \ \pi_{n_1+n_2-1,k_0}] = [0 \ 0 \ \dots \ 0] A^{k_0}$$

est **vecteur propre** de la matrice  $A^n$  associé à la **valeur propre**  $u$ .

Après un régime transitoire, pour  $k = 1, \dots, k_0$ , un **régime permanent** est atteint où, pour  $k \geq k_0$ , les dates  $\pi_{i,k}$ ,  $i = 1, \dots, n_1 + n_2 - 1$ , satisfont toutes

$$\pi_{i,k+n} = \pi_{i,k} \cdot u,$$

et ainsi les séquences de dates sont périodiques pour  $k \geq k_0$ , de même période  $n$ .

Prenant pour  $n$  la plus petite valeur possible, on a  $n = \text{ppcm}(n_1, n_2)$ .

Exploitant la commutativité de la multiplication, les dates  $\pi_{i,k}$  satisfont

$$\pi_{i,k+n} = \pi_{i,k} \cdot ((v_1 \cdot v_2 \ \dots \ v_{n_1})^{n/n_1} \vee (v_{n_1+1} \cdot v_{n_1+2} \ \dots \ v_{n_1+n_2})^{n/n_2}),$$

soit, en revenant à la notation  $(\max, +)$ ,

$$\begin{aligned} \pi_{i,k+n} &= \pi_{i,k} + \max\left(\frac{n}{n_1} \left(\sum_{i=1}^{n_1} v_i\right), \frac{n}{n_2} \left(\sum_{i=1}^{n_2} v_{n_1+i}\right)\right) \\ &= \pi_{i,k} + n \times \max\left(\frac{\sum_{i=1}^{n_1} v_i}{n_1}, \frac{\sum_{i=1}^{n_2} v_{n_1+i}}{n_2}\right) \end{aligned}$$

Notant  $\overline{v_1} = \frac{\sum_{i=1}^{n_1} v_i}{n_1}$  et  $\overline{v_2} = \frac{\sum_{i=1}^{n_2} v_{n_1+i}}{n_2}$  les valeurs moyennes des retards le long des circuits, l'incrément de dates associé à la période  $n$  est donc  $u = n \times \max(\overline{v_1}, \overline{v_2})$ .

On appelle  $\bar{v}_c = \frac{\sum_{i=1}^{n_c} v_i}{n_c}$  le temps moyen du cycle  $c$ . Dans un circuit  $c$ , chaque fois que l'on passe d'un nœud au suivant, on passe d'une itération à la suivante et la date est incrémentée de la durée associée à l'arc. Lorsque l'on fait un tour complet on retrouve la même variable, mais  $n_c$  itérations après, où  $n_c$  est le nombre d'arcs dans le circuit, et incrémentée d'une valeur  $u_c$  égale à la somme des valeurs des retards associées aux arcs du circuit. La valeur moyenne de l'incrément de date entre itérations consécutives le long du circuit est égale à  $\bar{v}_c = u_c/n_c$ , le temps moyen du cycle  $c$ . L'incrément des durées  $u$  pour la période  $n$  est  $n$  fois le temps moyen de cycle maximal.

Ce résultat se généralise à toute composante connexe  $C$ . Soit  $\mathcal{C}$  l'ensemble des circuits élémentaires de la composante connexe. La période  $n_C$  associée à la composante est le ppcm des nombres d'arcs dans les circuits,  $n_c$ , pris sur l'ensemble  $\mathcal{C}$  des circuits élémentaires  $c$ . L'incrément des durées pour la période  $n_C$  est

$$u_C = n_C \times \max_{c \in \mathcal{C}}(\bar{v}_c) = \text{ppcm}_{c \in \mathcal{C}}(n_c) \times \max_{c \in \mathcal{C}}(\bar{v}_c)$$

où  $\bar{v}_c = \frac{\sum_{i=1}^{n_c} v_i}{n_c}$  le temps moyen du cycle  $c$  et  $\max_{c \in \mathcal{C}}(\bar{v}_c)$  est le temps moyen de cycle maximal. La valeur  $\bar{v}_C = \max_{c \in \mathcal{C}}(\bar{v}_c)$  est appelée le temps moyen de cycle de la composante  $C$ .

Dans le cas général, le graphe du système a la structure d'un **graphe acyclique de composantes connexes**. Chaque composante a sa période propre  $n_C$  et sa valeur propre  $u_C$ , égale à  $n_C \times \overline{v}_C$  où  $\overline{v}_C$  est son temps moyen de cycle.

Lorsque l'ensemble du graphe est pris en compte, chaque composante atteint un régime périodique après un régime transitoire. Pour déterminer les périodes et valeurs propres associées, on applique récursivement la règle suivante : si la composante  $C$  est directement précédée par les composantes  $C_1, \dots, C_m$ , alors sa période est celle de la composante dont le temps moyen de cycle est maximale sur l'ensemble  $C = \{C_1, \dots, C_m, C\}$  et son temps moyen de cycle est égal à ce maximum.

La résolution complète du système d'équations récurrentes nécessite le calcul des régimes transitoires pour les composantes connexes. Alors que la période et le temps moyen de cycle de chaque composante peuvent être calculées efficacement, sans énumérer ses cycles élémentaires, il n'en est pas de même pour le calcul des dates pendant les régimes transitoires ( $k < k_0$ ,  $k_0$  dépendant de la composante); il faut appliquer les équations du système et dérouler les calculs.

**Remarque** : les références sur les algorithmes efficaces pour le calcul du temps moyen de cycle maximal associé à une composante connexe se trouvent facilement avec les mots clés *maximum mean cycle time*.