

# C3: SYNTAXE

- LE MODELE GRAFCET
- LES ETAPES
- LES ACTIONS ASSOCIEES AUX ETAPES
  - Les actions continues
  - Les actions mémorisées
  - Commentaires
- LES TRANSITIONS
  - Définition et symbole
  - Exemples
  - Transition validée
- LES RECEPTIVITES
  - Définition
  - Réceptivité toujours vraie
  - Réceptivité sur front
  - Réceptivité liée au temps
  - Réceptivité liée avec la valeur booléenne d'un prédicat
- LIAISONS ORIENTEES
  - Définition
  - Règle de tracé
  - Utilisation de renvois dans les liaisons
- REGLE DE SYNTAXE

Le GRAFCET est régie par la norme **NF EN 60848 d'Août 2002.**

**GRAFCET: GR**Aphe **Fonctionnel** de **Commande** **Etape**  
**Transition**

Le langage de spécification **GRAFCET** est un modèle de représentation graphique du comportement de la partie commande d'un système automatisé.

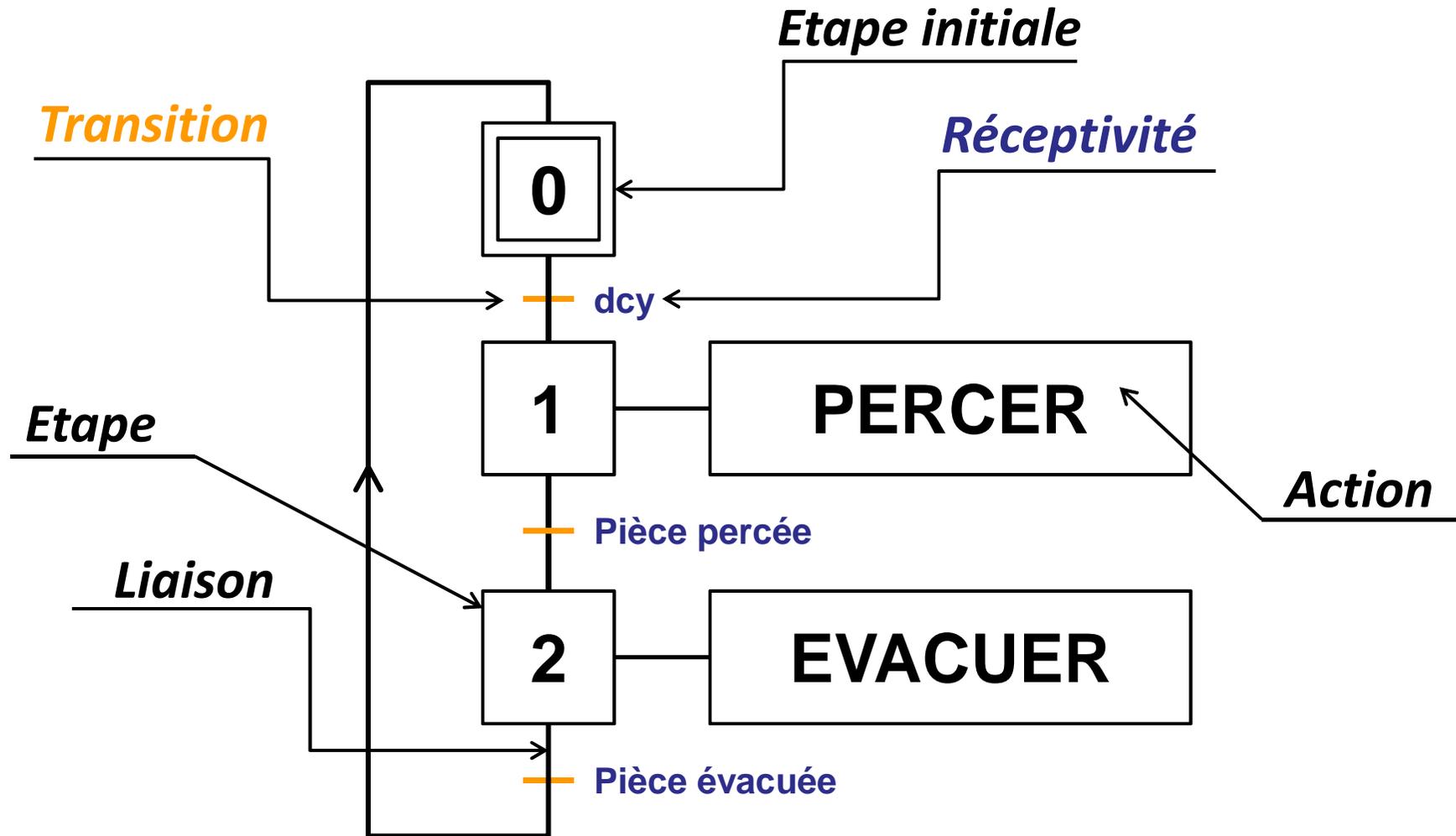


### Ce modèle est défini par un ensemble constitué :

- **d'éléments graphiques de base** comprenant : les étapes, les transitions et les liaisons;
- **d'une interprétation** traduisant le comportement de la partie commande vis-à-vis de ses entrées et de ses sorties, et caractérisée par les réceptivités associées aux transitions et les actions associées aux étapes ;
- **de 5 règles d'évolution** définissant formellement le comportement de la partie commande ;
- **de postulats sur les durées** relatives aux évolutions,

# LE MODELE GRAFCET

## Eléments graphiques de base

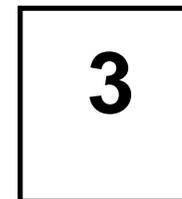


**Une ETAPE caractérise le comportement invariant du système.**

**Elle se représente par un carré repéré numériquement (deux étapes d'un même grafcet ne peuvent porter le même numéro) :**

**A un instant donné et suivant l'évolution du système, une étape est:**

soit inactive,



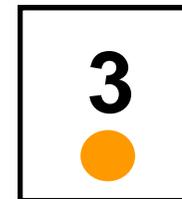
**Une ETAPE caractérise le comportement invariant du système.**

**Elle se représente par un carré repéré numériquement (deux étapes d'un même grafcet ne peuvent porter le même numéro) :**

**A un instant donné et suivant l'évolution du système, une étape est:**

**soit inactive,**

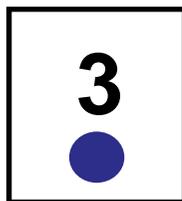
**soit active.**



On peut associer une variable à l'état d'une étape.  
Cette variable, appelée **variable d'étape**, est notée  $X_i$ , elle prend la valeur:

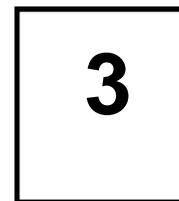
- 1** lorsque l'étape  $i$  est active
- 0** lorsque l'étape  $i$  est inactive

Etape 3 active



$$X3 = 1$$

Etape 3 inactive



$$X3 = 0$$

Une **SITUATION D'UN GRAFCET** est défini par l'ensemble des étapes actives à un instant donné.

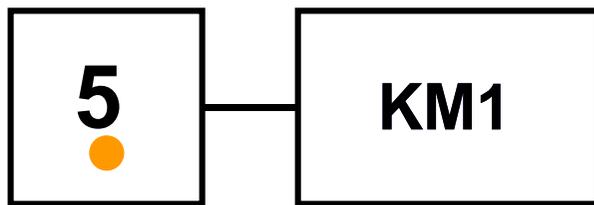
Exemple :  $S = \{1, 14, 16\}$  ou  $S(Gm, Gn, Gp) = \{1, 14, 16\}$

- « **Situation vide** » → aucune étape n'est active.  
Elle se note  $S = \{ \}$ .
- « **Situation courante** » → à un moment donné.  
Elle se note  $S = \{ * \}$ .
- « **Situation initiale** » → certaines étapes sont initialement activées. Elles sont repérées en doublant le cadre symbolique de l'étape.

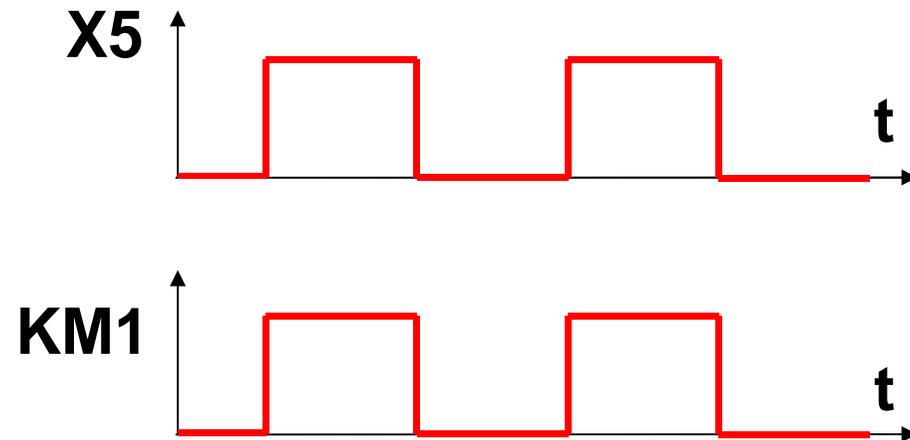
- Les **actions** permettent d'établir le lien entre l'évolution du grafcet et les sorties.
- Deux modes, ***mode continu*** ou ***mode mémorisé***, décrivent comment les sorties dépendent de l'évolution et des entrées du système.

### Les Actions continues inconditionnelles

L'**action continue**, associée à une étape, dure tant que l'étape est active si aucune condition d'assignation ne l'interdit.



On note:  $KM1 = X5$



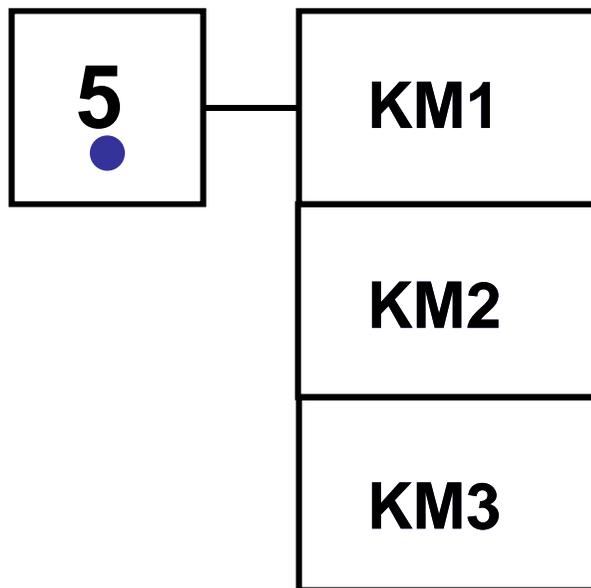
La **condition d'assignation** est une expression logique de variables d'entrée et/ou de variables internes

**Assignation** : le fait d'imposer la valeur (vraie ou fausse) des variables de sortie.

# LE MODELE GRAFCET

## Les actions continues

Une ou plusieurs actions peuvent être associées à une étape.  
 Cette représentation n'implique pas de priorité entre ces actions.

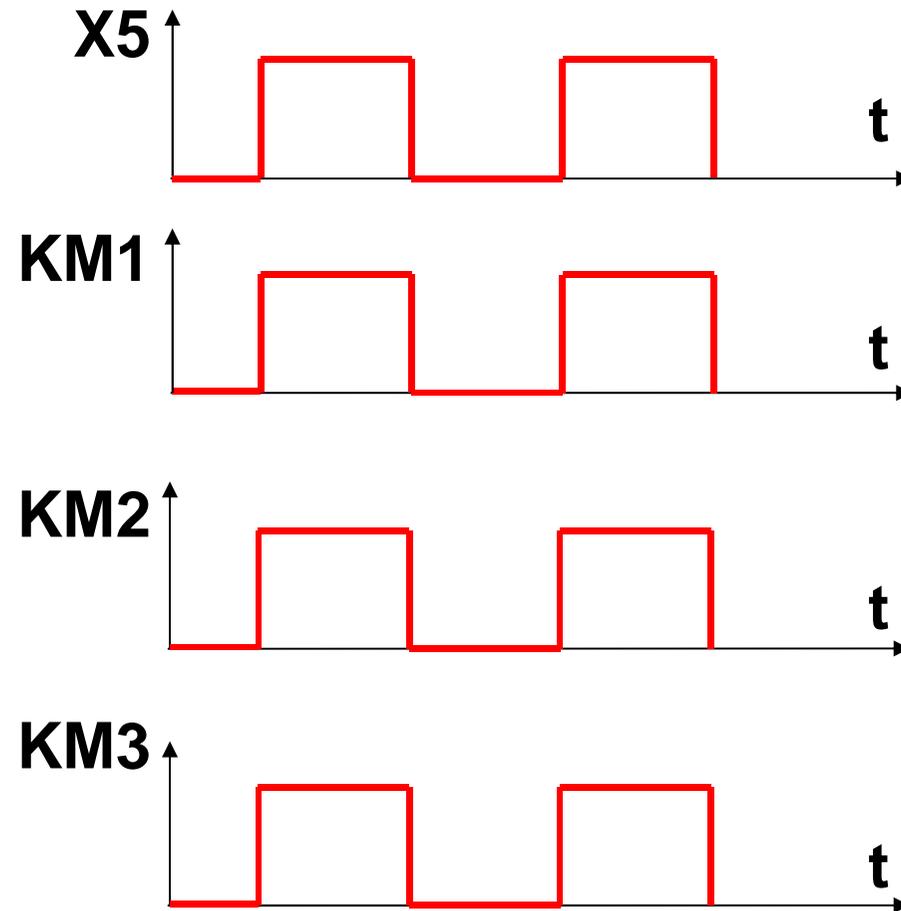


On note:

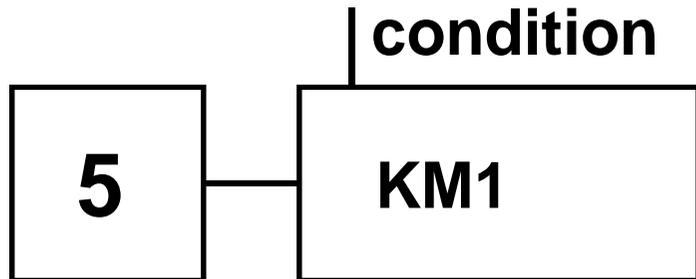
$$KM1 = X5$$

$$KM2 = X5$$

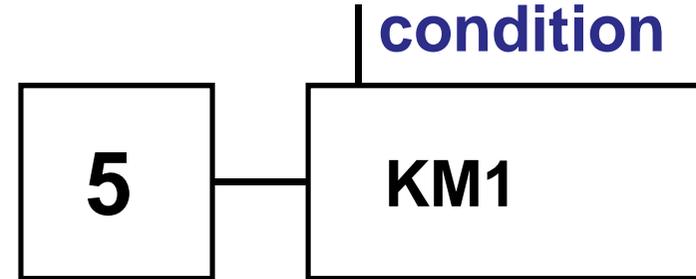
$$KM3 = X5$$



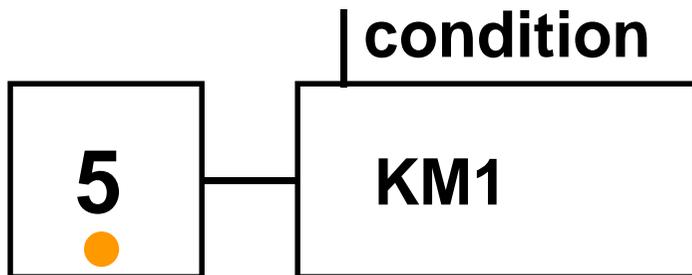
### b. Actions conditionnelles



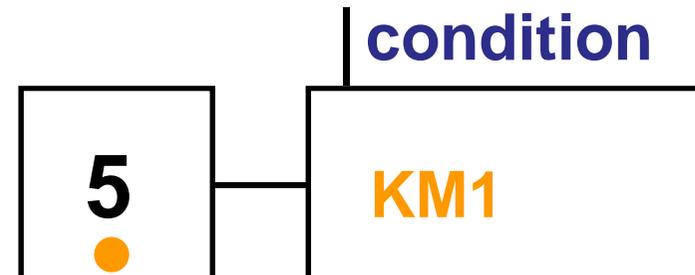
L'action KM1 n'est pas effectuée



L'action KM1 n'est pas effectuée



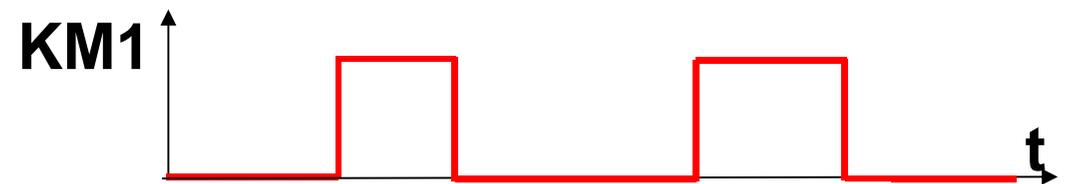
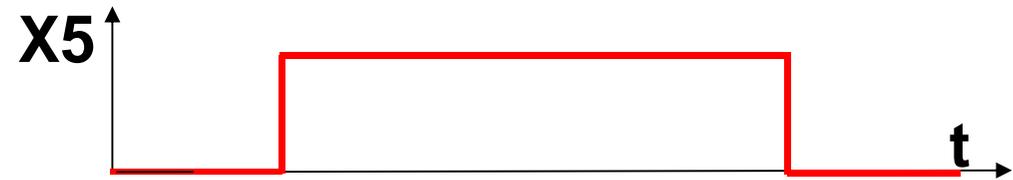
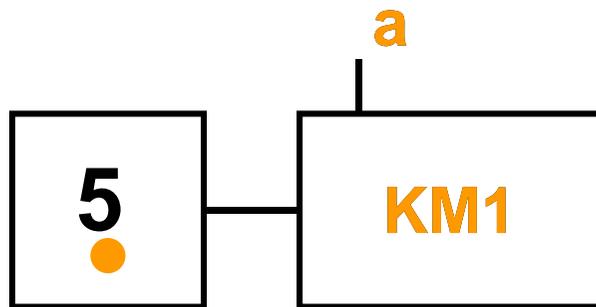
L'action KM1 n'est pas effectuée



L'action **KM1** est effectuée  
On note: **KM1 = X5.condition**

# 1. Les actions continues

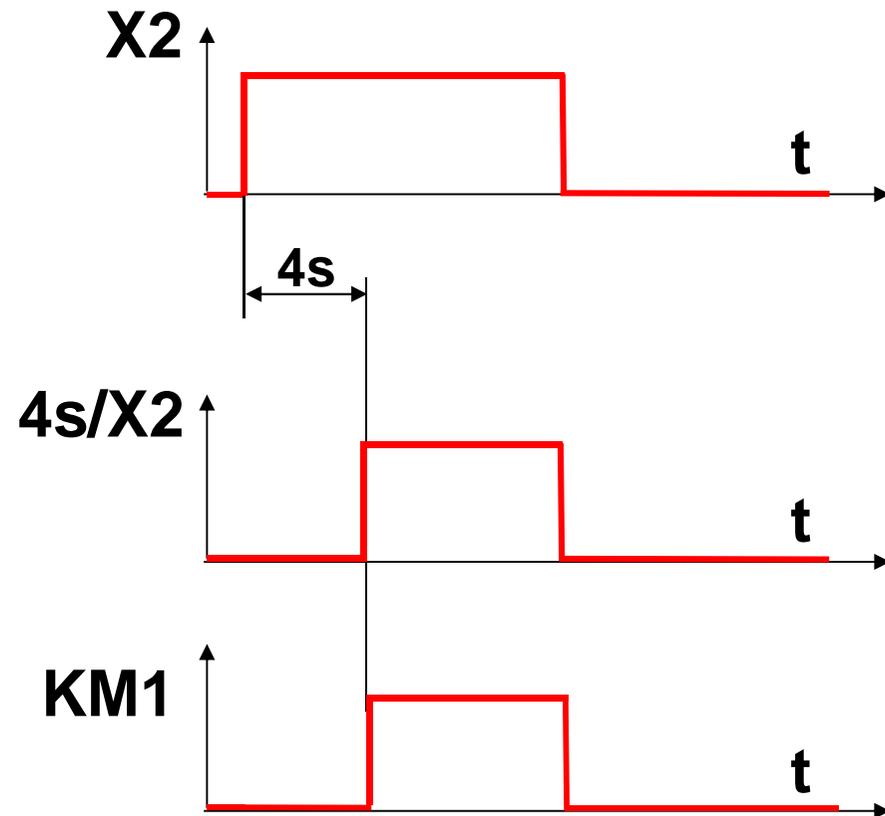
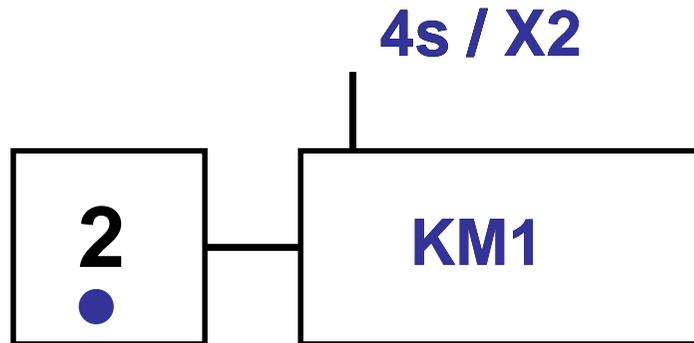
## b. Actions conditionnelles



La **condition d'assignation** ne doit jamais comporter de **front de variable**

# 1. Les actions continues

## c. Actions retardées



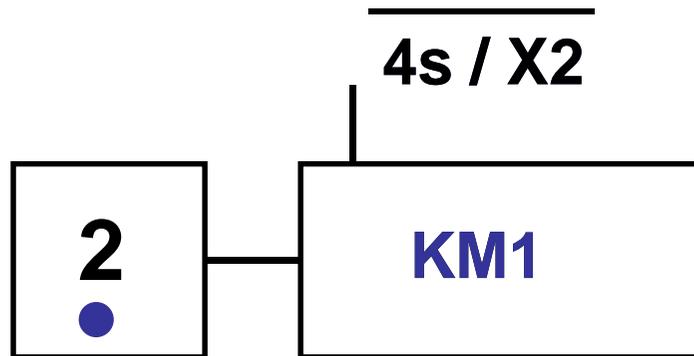
On note:  $KM1 = 4s / X2$

*Si la durée d'activité de l'étape 2 est inférieure à 4s, la sortie KM1 ne sera pas assignée à la valeur vraie.*

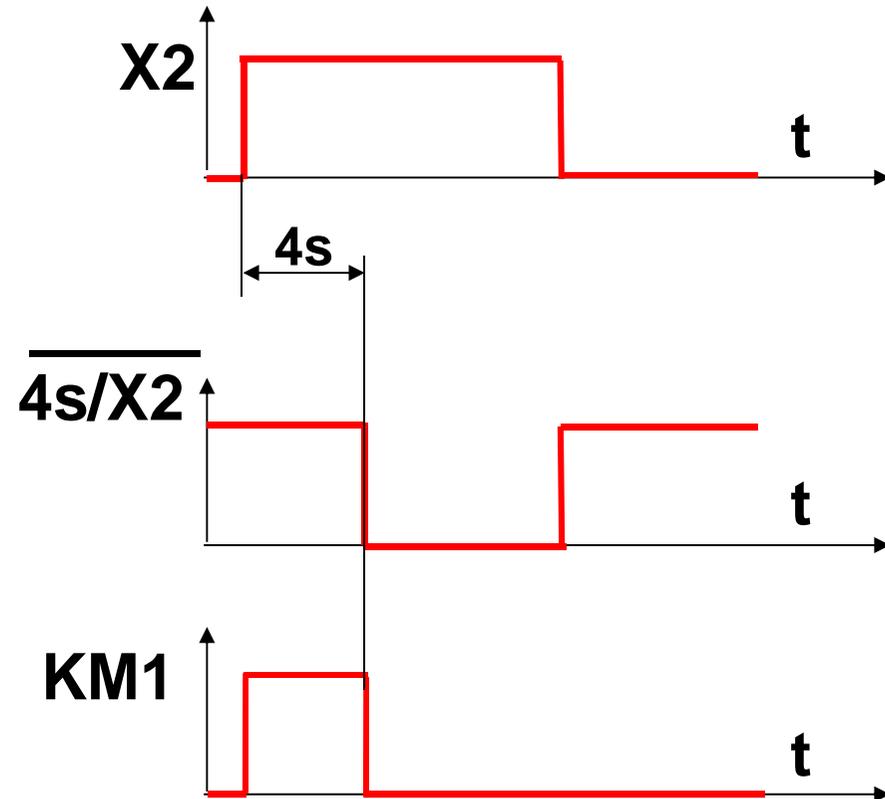


# 1. Les actions continues

## c. Actions limitées dans le temps



On note:  $KM1 = X2 \cdot \overline{4s / X2}$



L'action *KM1* sera exécutée pendant **au plus 4s** dès l'activation de l'étape 2.

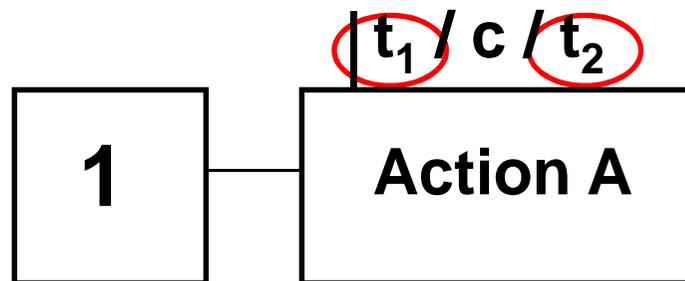


# 1. Les actions continues

## *d. Actions à activation et désactivation retardées*

La notation  $t_1 / c / t_2$  indique que la **condition d'assignation** n'est vraie qu'après un temps  $t_1$  depuis le front montant de  $c$  et redevient fausse après un temps  $t_2$  depuis le front descendant de  $c$ .

**$t_1$  et  $t_2$  doivent être remplacés par leur valeur réelle**

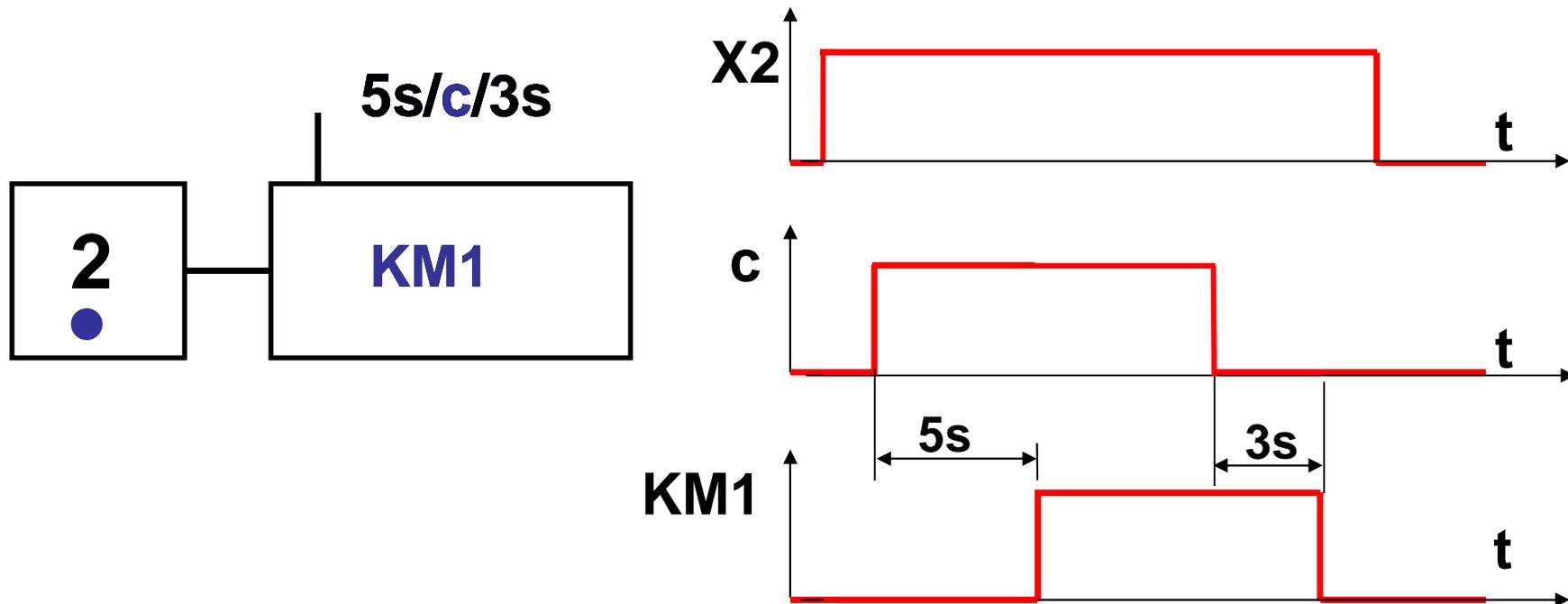


**La variable  $c$  doit rester vraie un temps supérieur ou égal à  $t_1$  pour que la condition d'assignation puisse être vraie**



# 1. Les actions continues

## e. Actions à activation et désactivation retardées



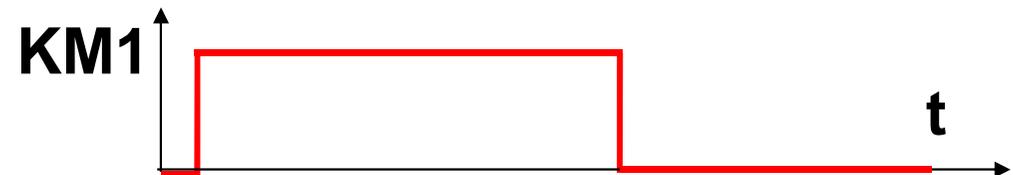
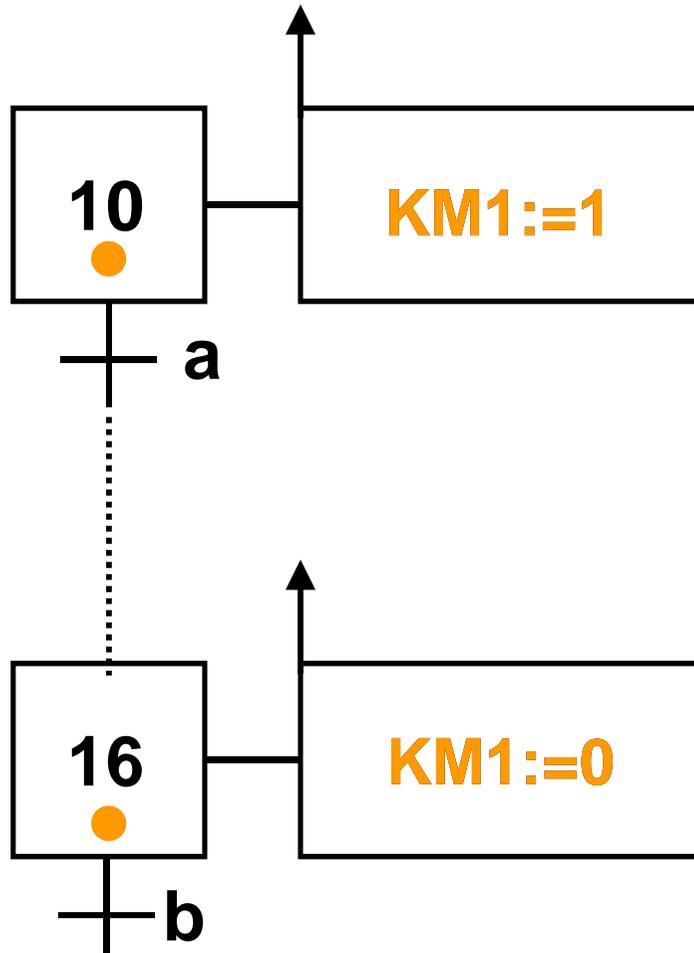
On note:  $KM1 = X2. 5s/c/3s$

Si  $c$  passe à 0 avant 5s, la **condition d'assignation** ne pourra être vraie : il faudra attendre le prochain front montant de  $c$ .



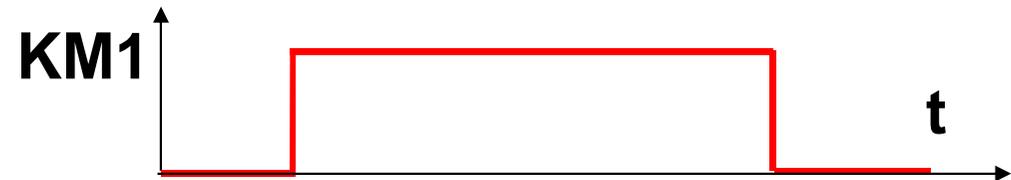
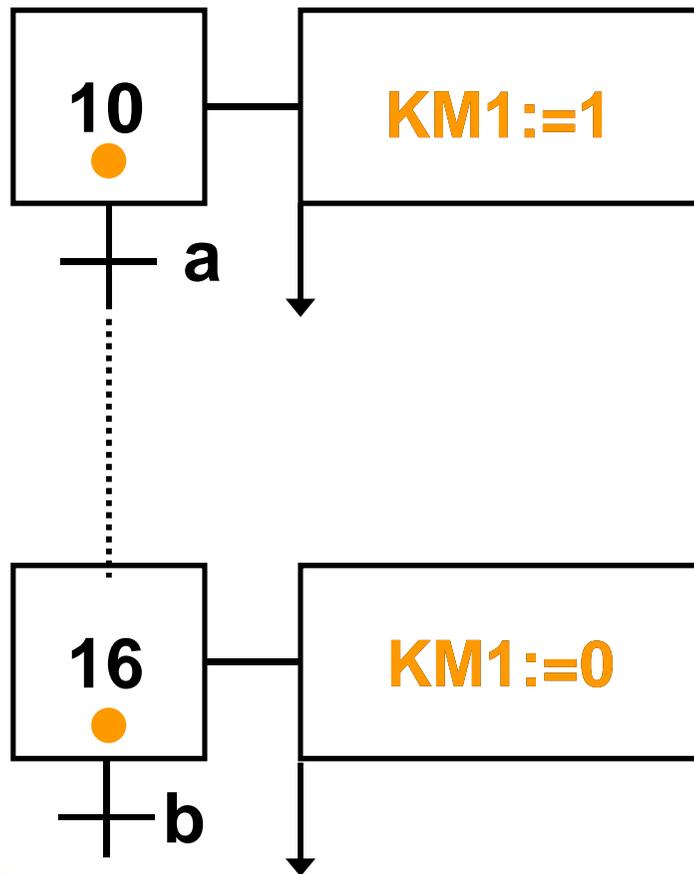
## 2. Les actions mémorisées

### a. Action à l'activation



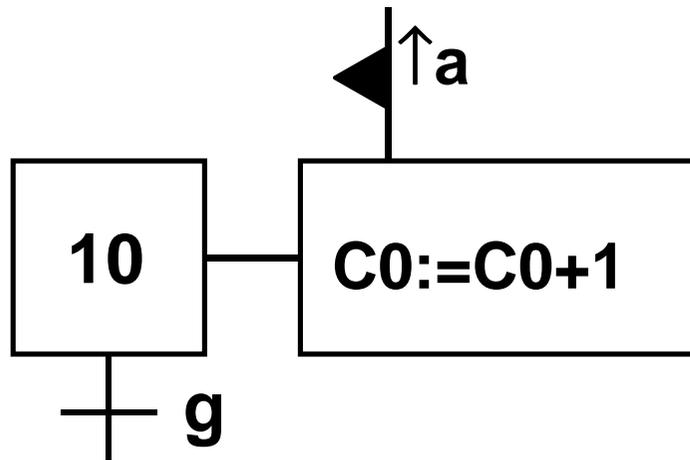
## 2. Les actions mémorisées

### *b. Action à la désactivation*



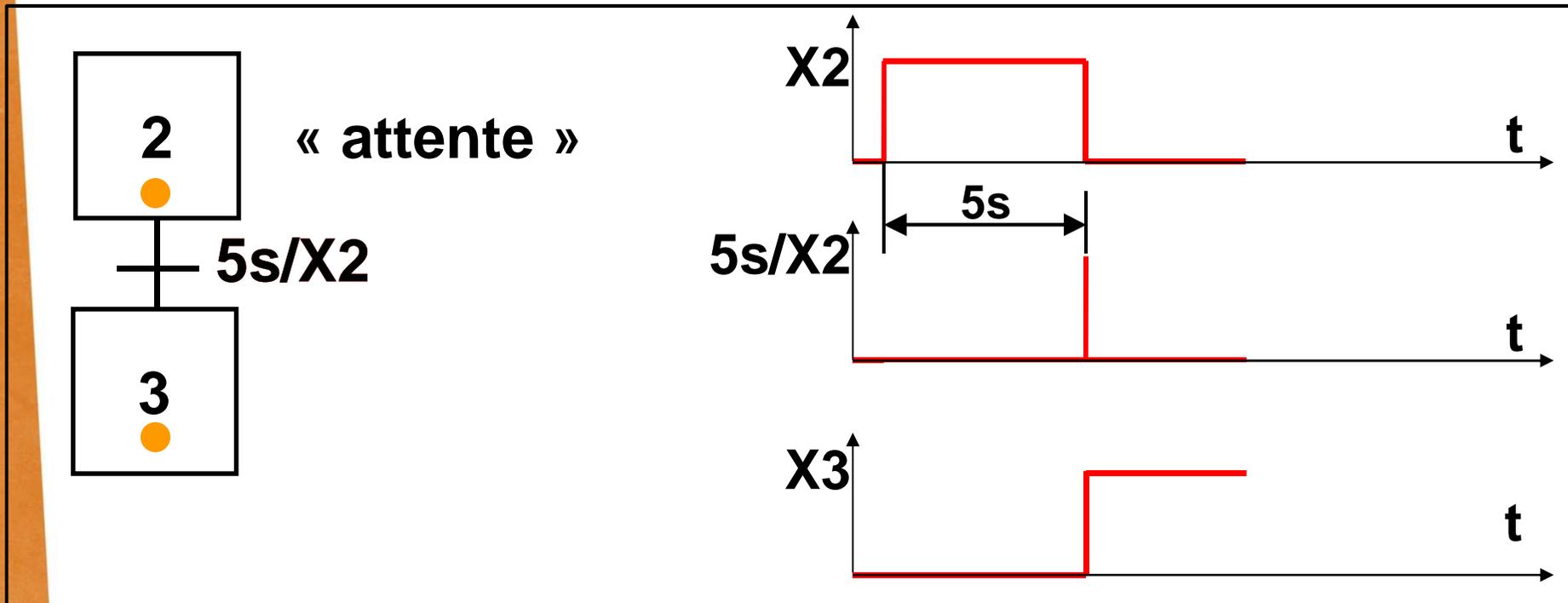
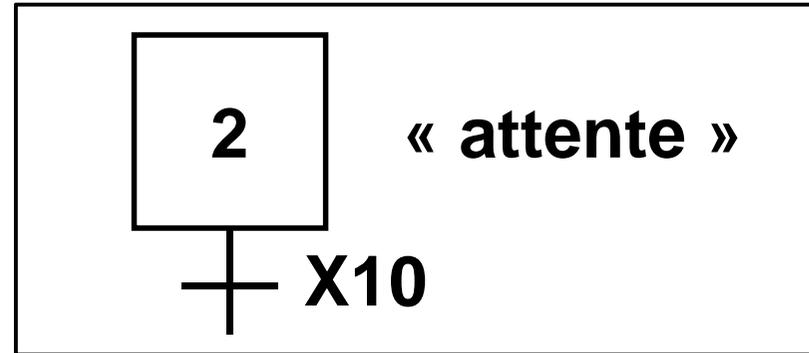
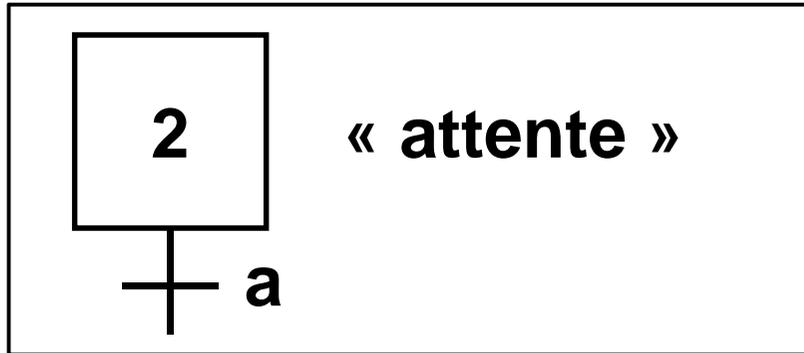
## 2. Les actions mémorisées

### c. Action sur événement



La notation de l'affectation s'écrit avec le signe `:=` ; cette notation s'applique tout autant aux affectations booléennes « pousser pièce:=1 » qu'aux affectations numériques « `C :=C+1` ».

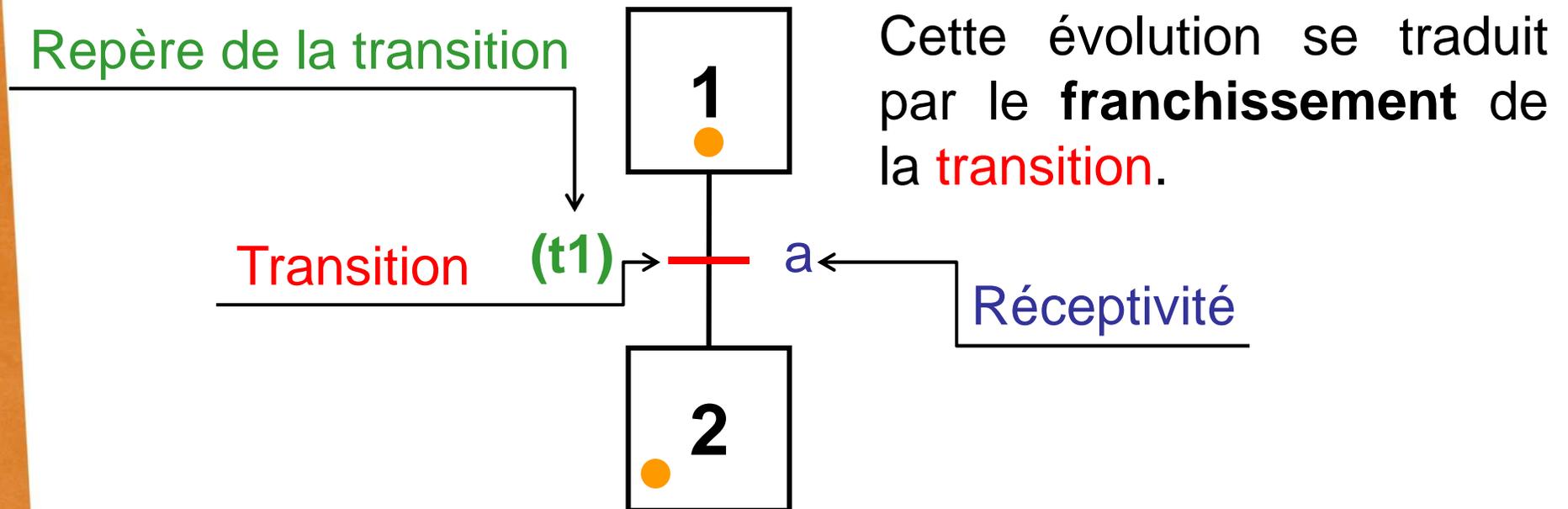
### 3. Commentaires



# IV. TRANSITIONS

## 1. Définition et symbole

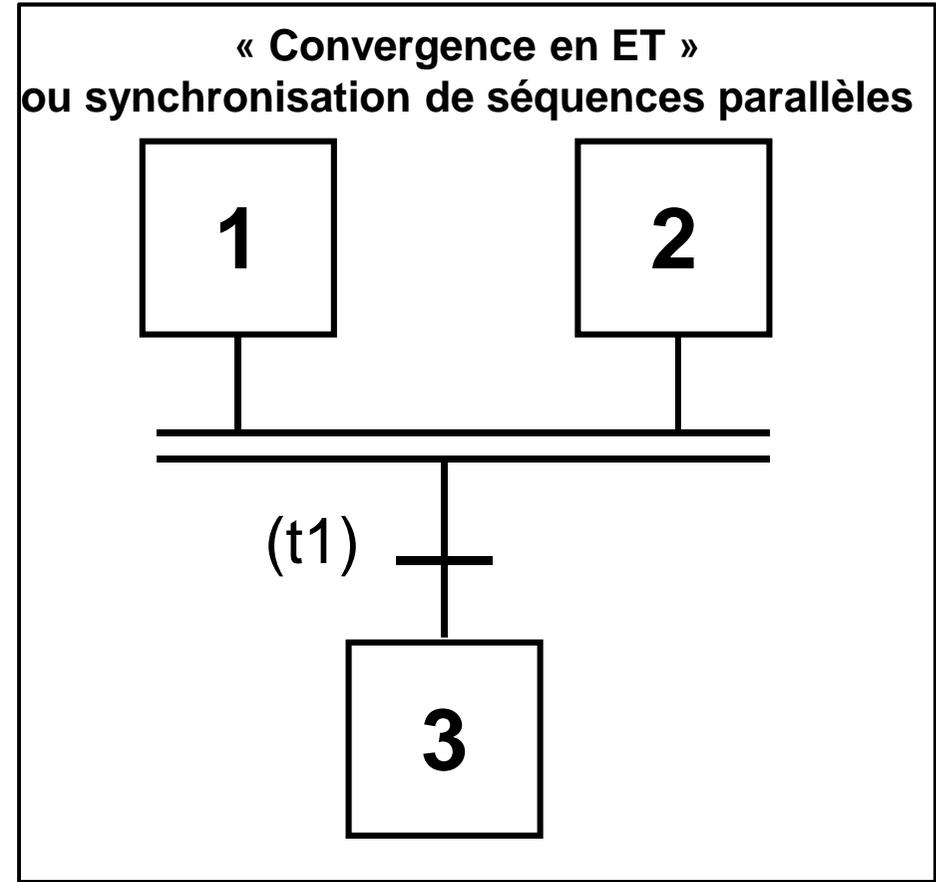
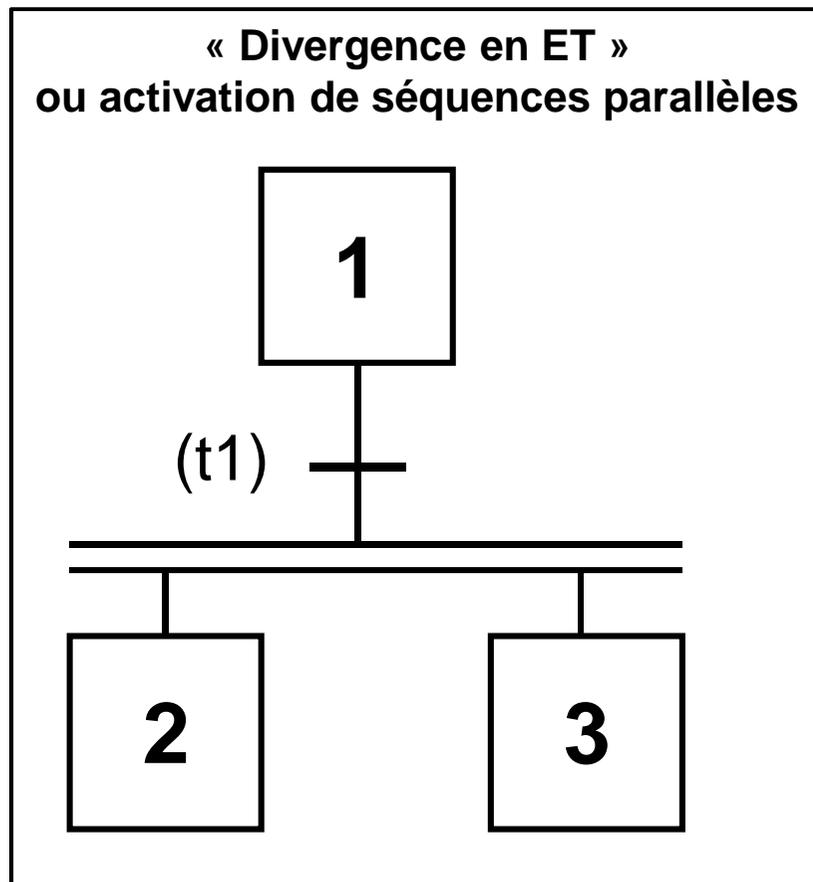
Une **transition** indique la possibilité d'évolution d'une (ou plusieurs) étape(s) active(s) vers une ou plusieurs autre(s) étape(s).



# IV. TRANSITIONS

## 2. Exemples

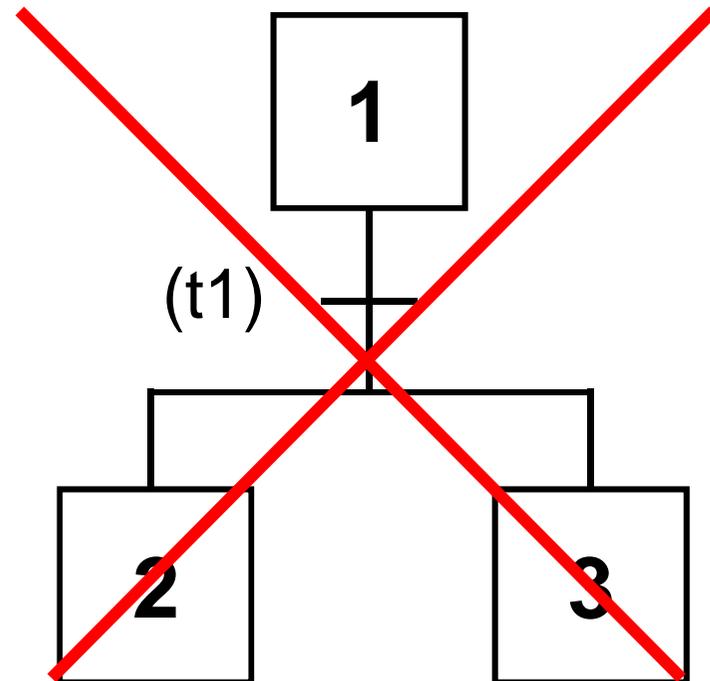
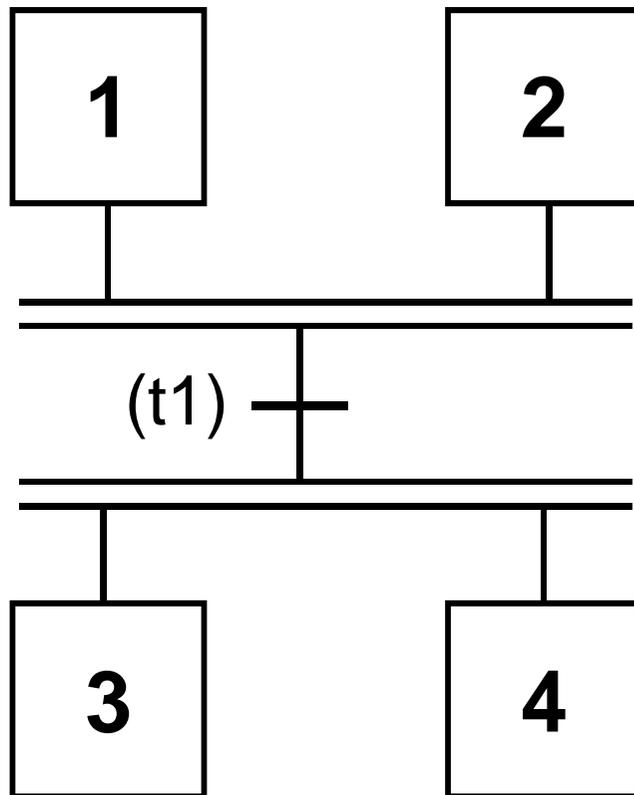
Lorsque plusieurs étapes sont reliées à une transition commune, elles sont toutes obligatoirement reliées à une double barre appelée **barre de synchronisation**.



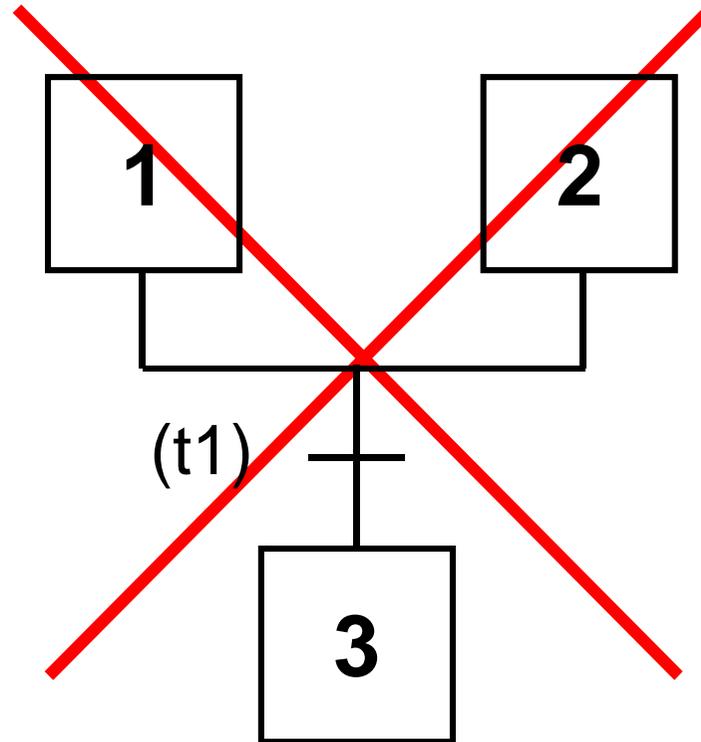
# IV. TRANSITIONS

## 2. Exemples

« Synchronisation » et « activation de séquences parallèles »



## 2. Exemples

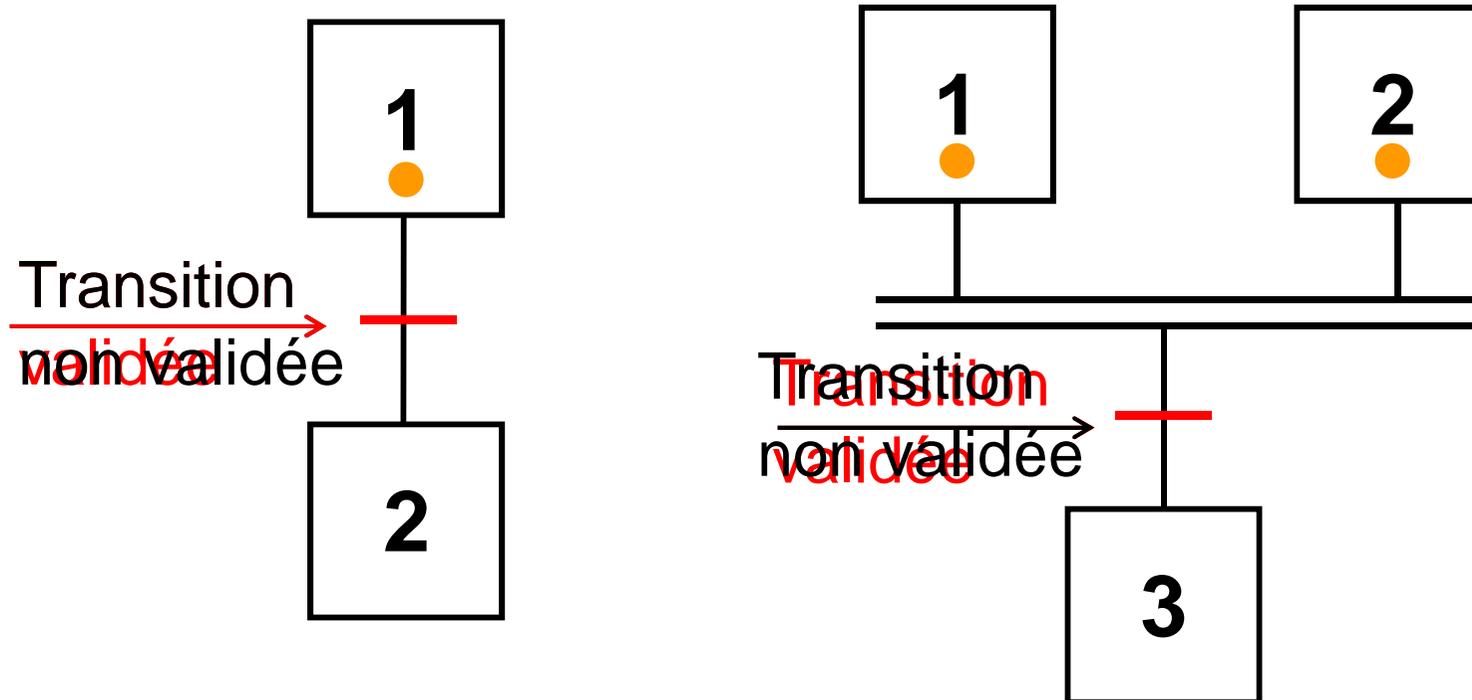


# IV. TRANSITIONS

## 2. Transition validée

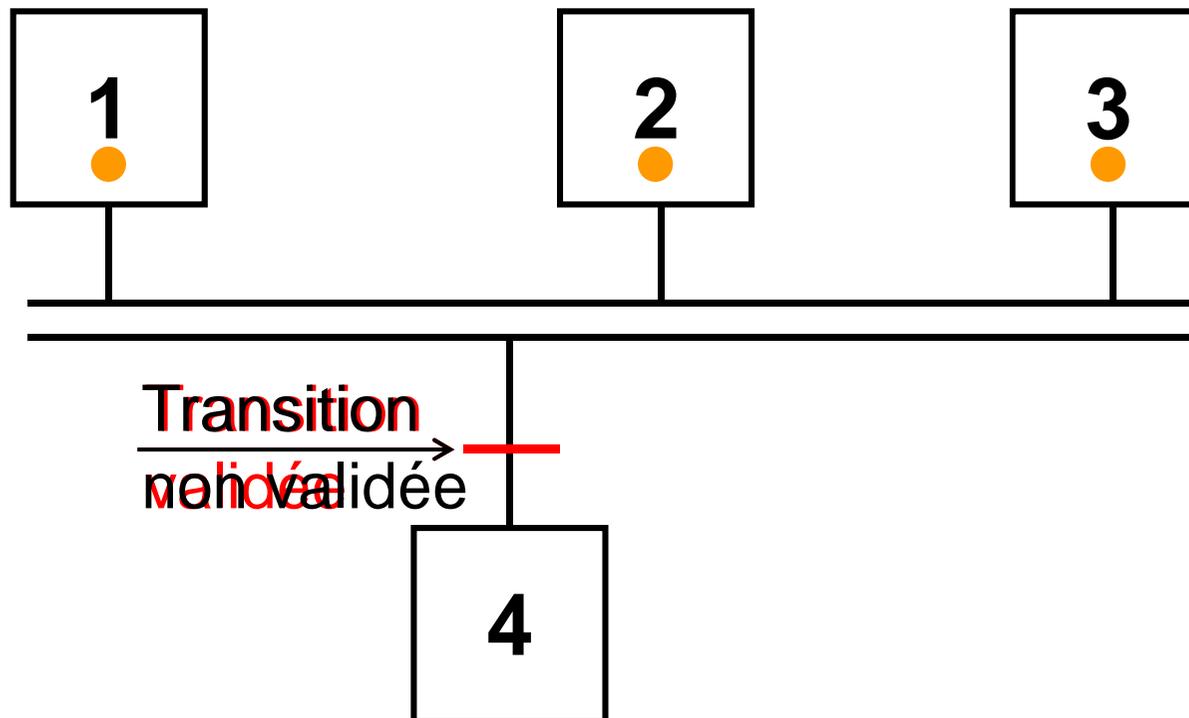
Une transition est soit **validée**, soit **non validée**.

Elle est validée **lorsque toutes les étapes précédentes sont actives**.



# IV. TRANSITIONS

## 3. Transition validée



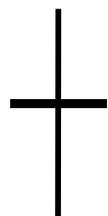
# V. RECEPTIVITES

## 1. Définition

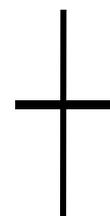
Une proposition logique, appelée **réceptivité**, qui peut être soit **vraie** soit **fausse**, est associée à chaque transition.

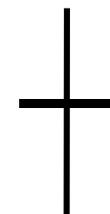
*Description d'une réceptivité par:*

*une expression littérale*


 Présence pièce

*une expression logique*


 a. (b+/c) Variables d'entrées


 X1+X2 Variables internes

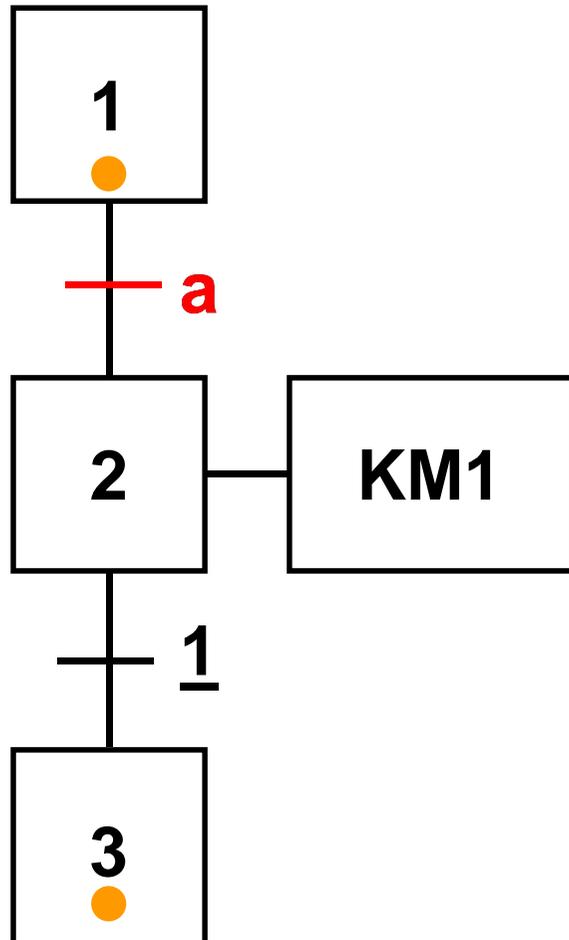
## 2. Réceptivité toujours vraie

La notation 1 indique que la réceptivité est toujours vraie.

Dans ce cas, l'évolution est dite **fugace**, le franchissement de la transition n'est conditionné que par l'activité de l'étape amont.

# V. RECEPTIVITES

## □ Evolution fugace

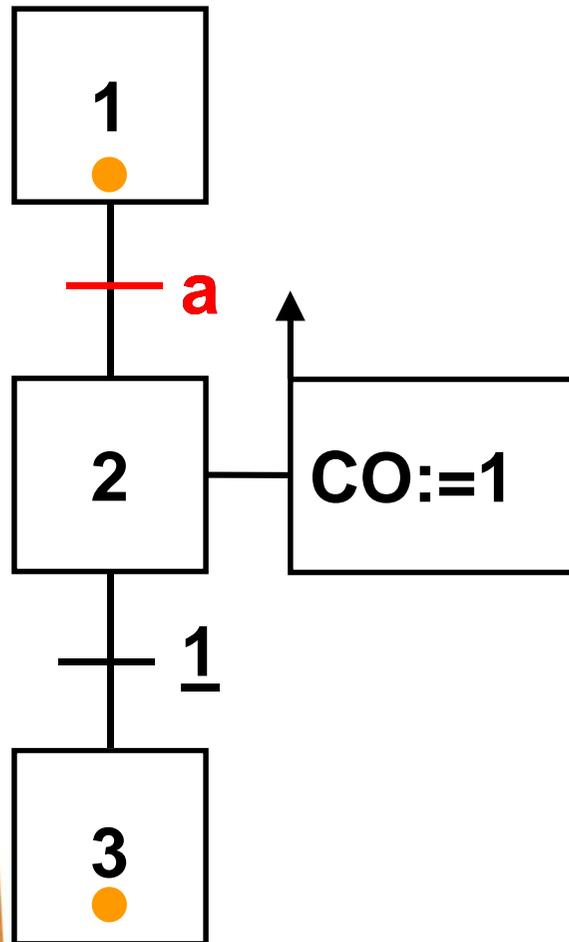


L'étape 2 est instable.

Une action continue associée à une étape instable n'est pas assignée à la valeur 1 : **KM1 ne sera pas vraie.**

# V. RECEPTIVITES

## □ Evolution fugace



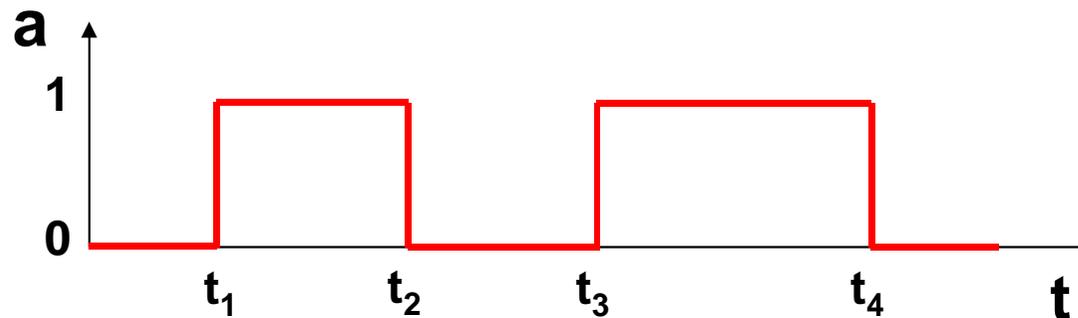
L'étape 2 est instable.

Une action mémorisée associée à une étape instable permet d'assigner la valeur à la variable : **la valeur du compteur C est égale à 1.**

## 3. Réceptivité sur front

### □ *Événement d'entrée*

Un **événement d'entrée** est un **changement d'état** d'une variable d'entrée associé à la date à laquelle il se produit.



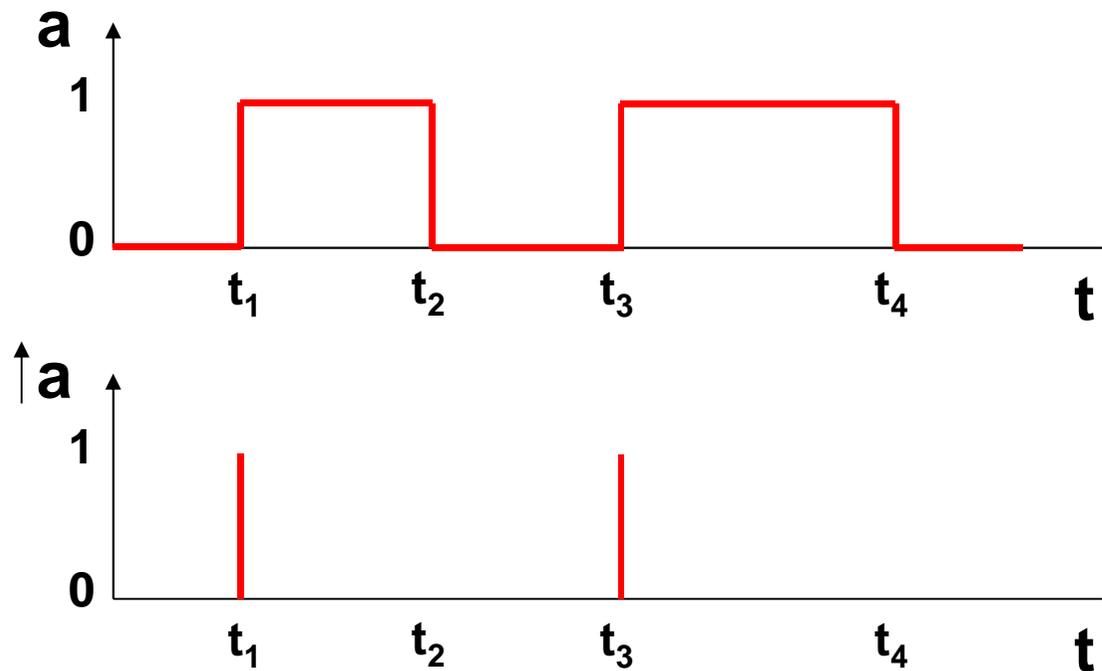
Des événements se produisent aux temps  $t_1$ ,  $t_2$ ,  $t_3$  et  $t_4$  lorsque la variable  $a$  change d'état.

# V. RECEPTIVITES

## 3. Réceptivité sur front

On distingue deux types d'événement:

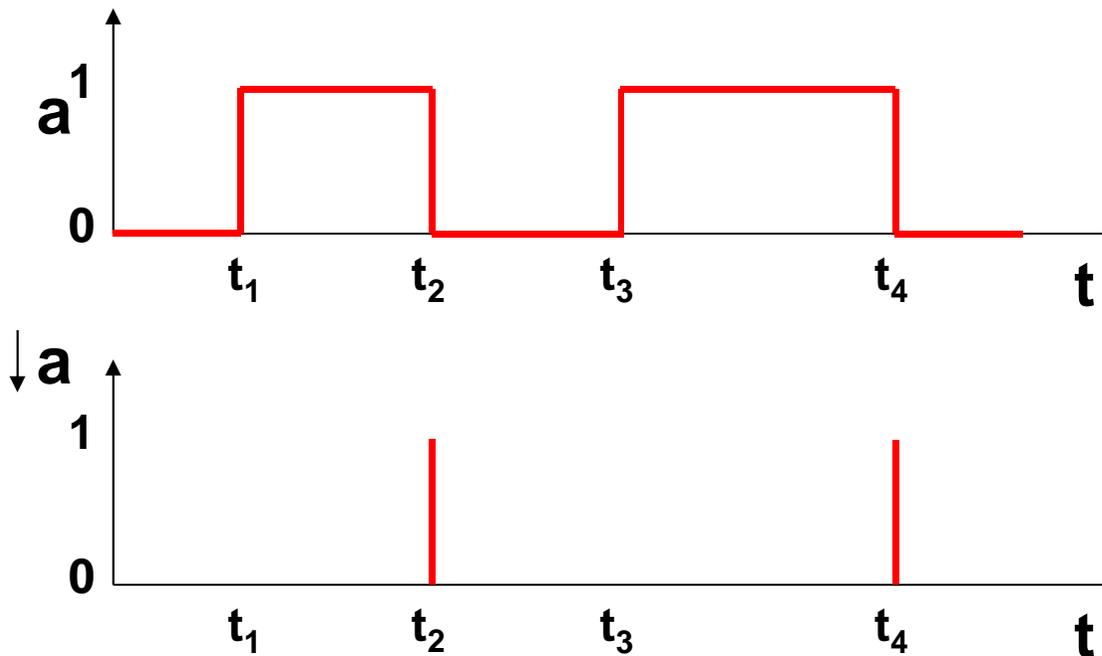
- le **front montant** de  $a$ , noté  $\uparrow a$ , qui ne prend la valeur 1 qu'au passage de la valeur 0 à 1 de la variable  $a$ .



# V. RECEPTIVITES

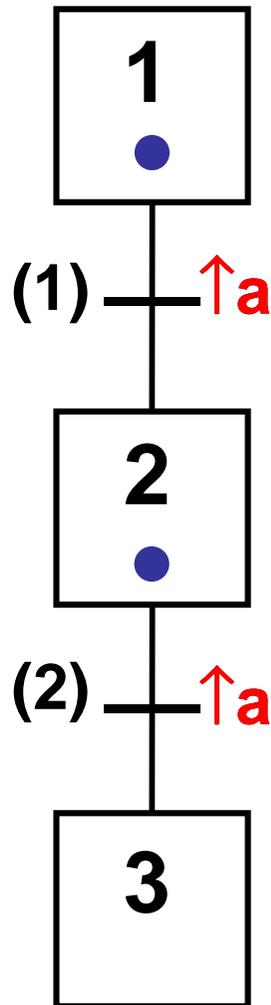
## 3. Réceptivité sur front

- le **front descendant** de  $a$ , noté  $\downarrow a$ , qui ne prend la valeur 1 qu'au passage de la valeur 1 à 0 de la variable  $a$ .

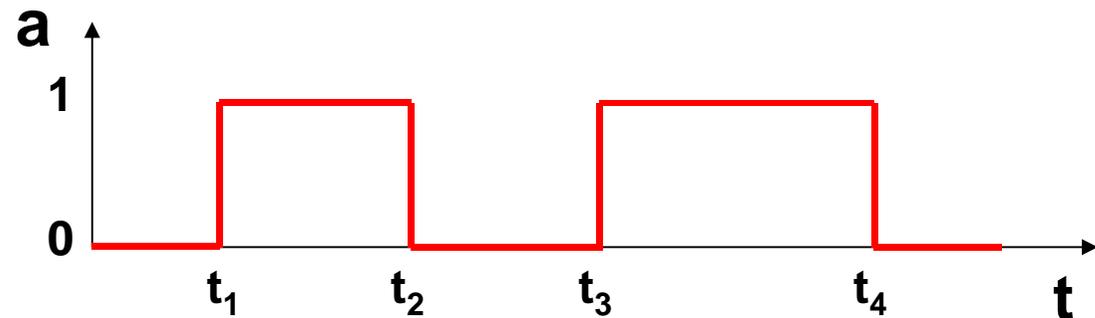


# V. RECEPTIVITES

## □ Exemple



## 3. Réceptivité sur front



Plaçons nous à l'instant  $t_1$

Seule la transition (1) est franchie.

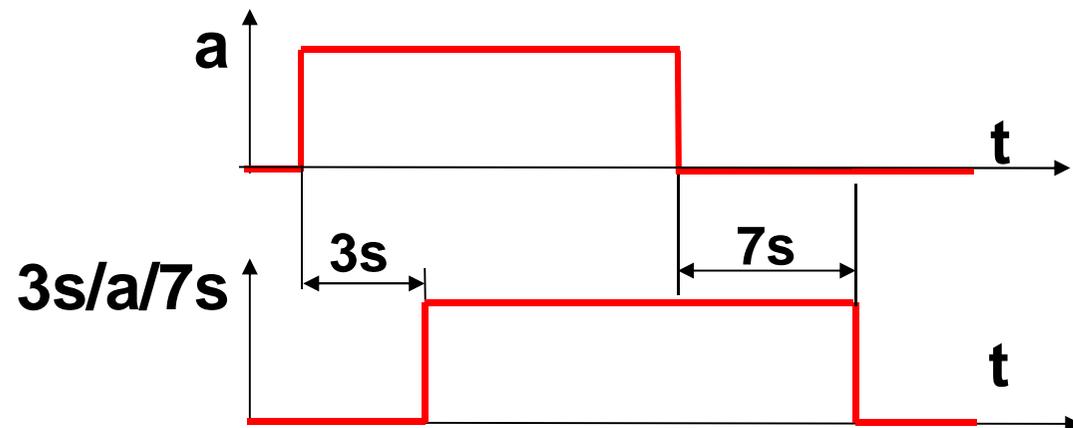
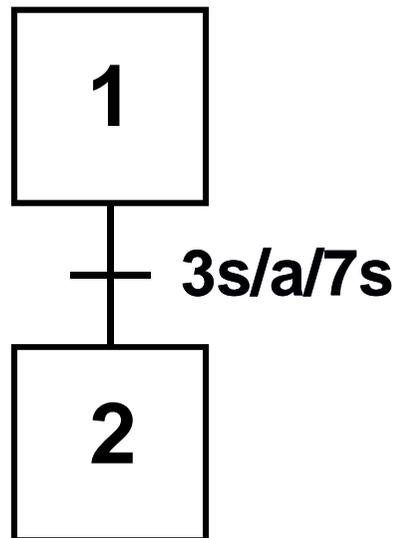
La transition (2) est validée, mais la réceptivité  $\uparrow a$  n'est plus vraie, il faudra attendre  $t_3$  pour la franchir.



# V. RECEPTIVITES

## 4. Réceptivité liée au temps

La notation est de la forme «  $t_1/\text{variable}/t_2$  ».

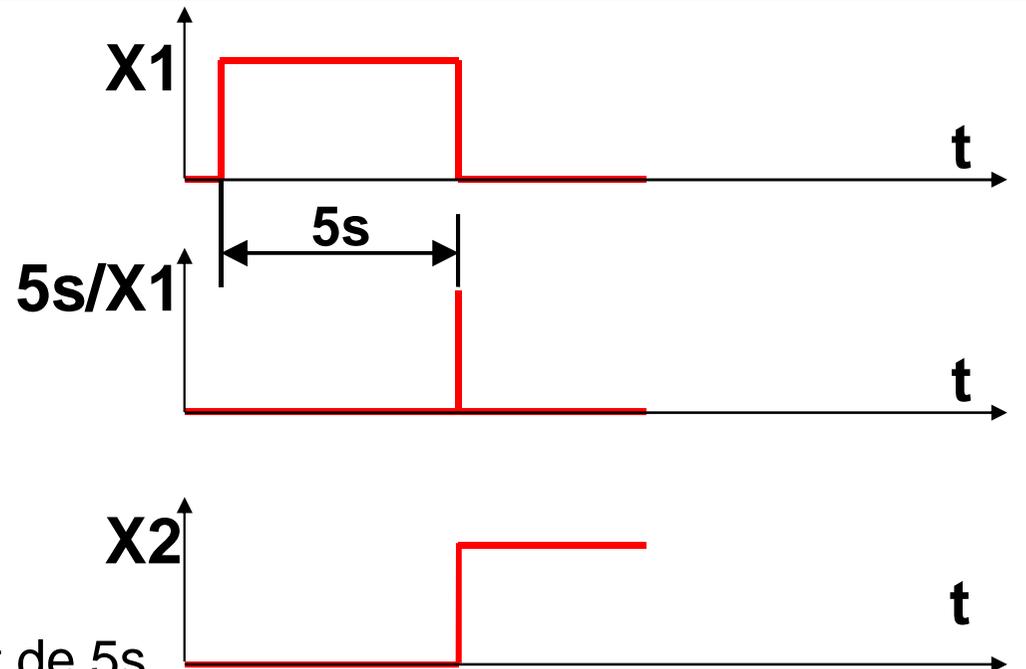
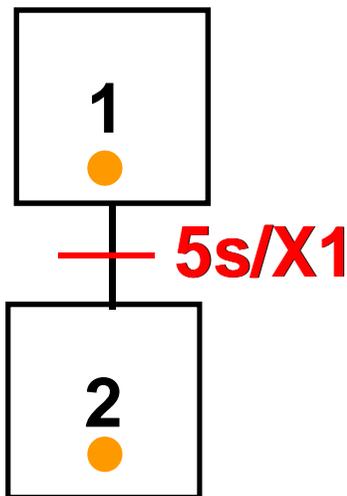


Remarque: La variable temporisée, ici  $a$ , doit rester vraie pendant un temps égal au supérieur à  $t_1$ , soit ici  $3s$ , pour que la réceptivité puisse être vraie.

## 4. Réceptivité liée au temps

### □ *Simplification usuelle*

L'utilisation la plus courante est la temporisation **de la variable d'étape avec un temps  $t_2$  égal à 0.**

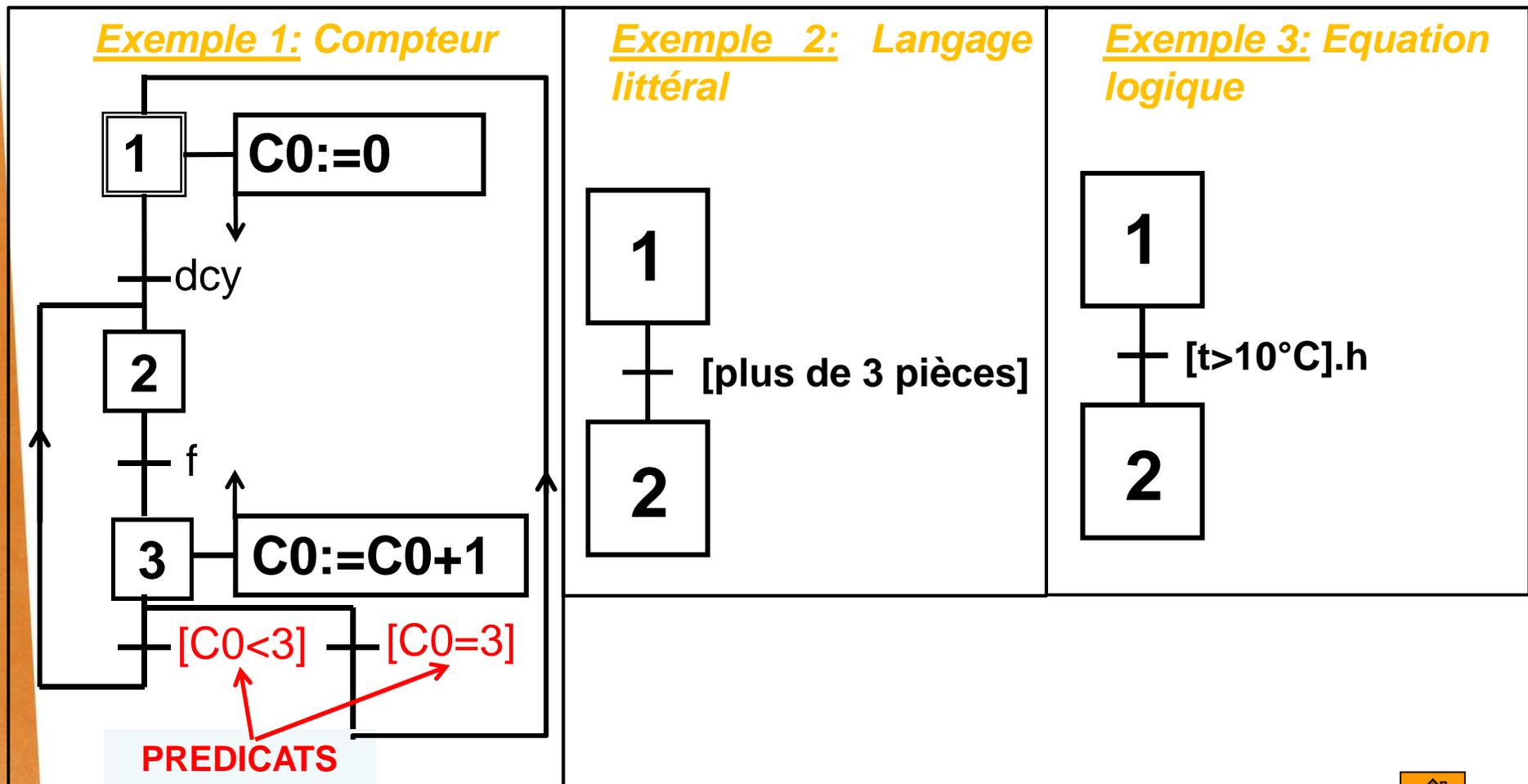


La durée d'activité de l'étape 1 est de 5s.

# V. RECEPTIVITES

## 5. Réceptivité avec la valeur booléenne d'un prédicat

Un *prédicat* est une expression contenant une ou plusieurs variables et qui est susceptible de devenir une proposition vraie ou fausse.



# VI. LIAISONS ORIENTÉES

## 1. Définition

Les liaisons orientées relient les étapes aux transitions et les transitions aux étapes. Elles indiquent les voies d'évolution.

## 2. Règles de tracé



↓ sens de lecture

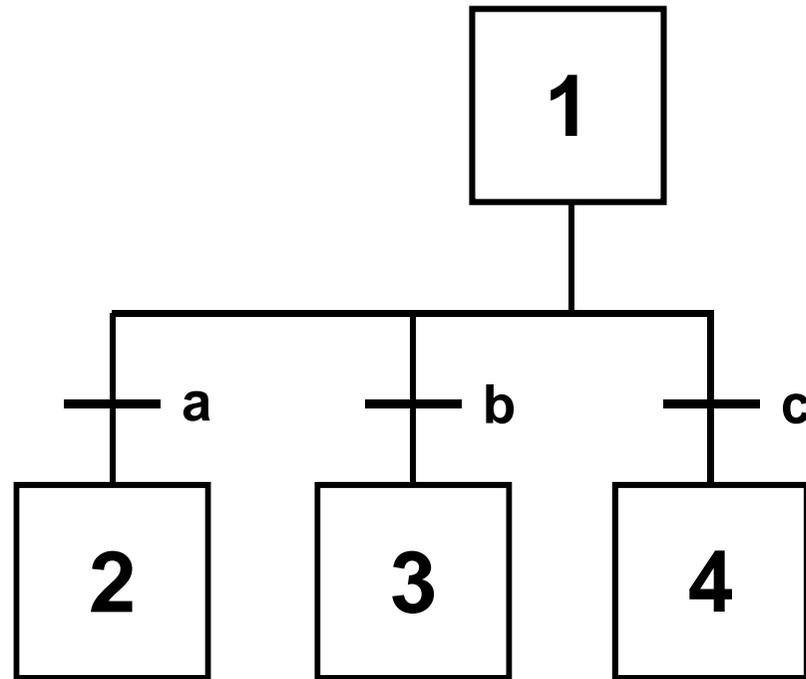
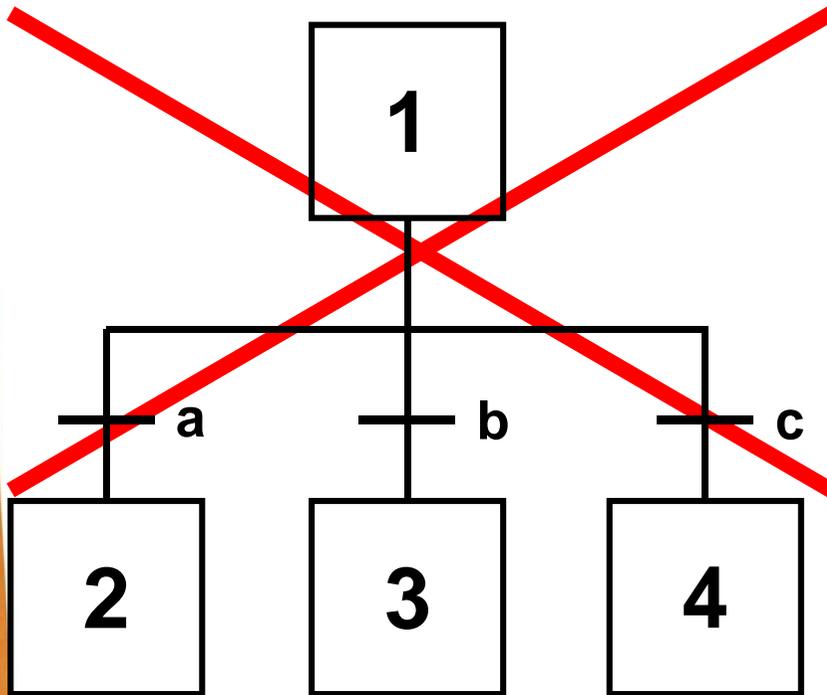


↑ sens de lecture

# VI. LIAISONS ORIENTÉES

## 2. Règles de tracé

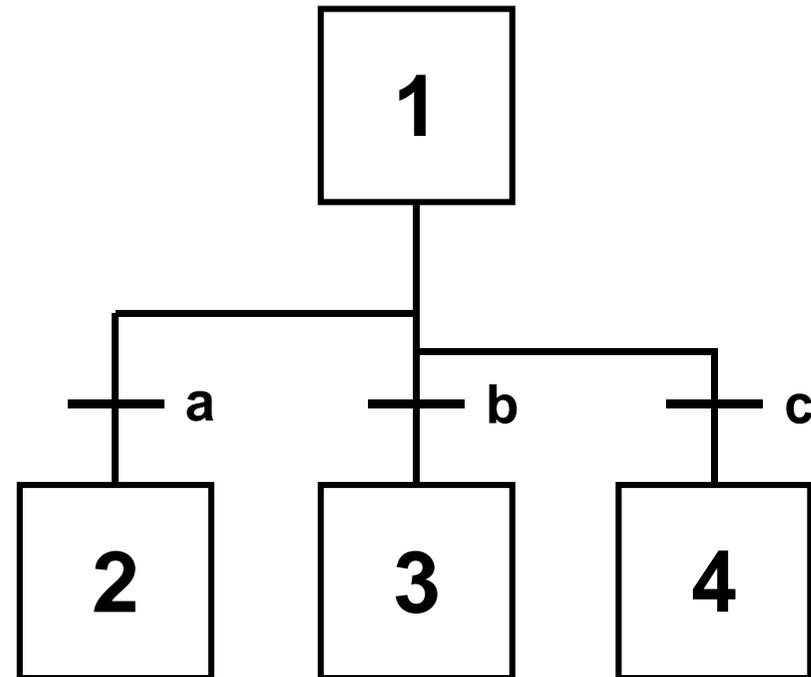
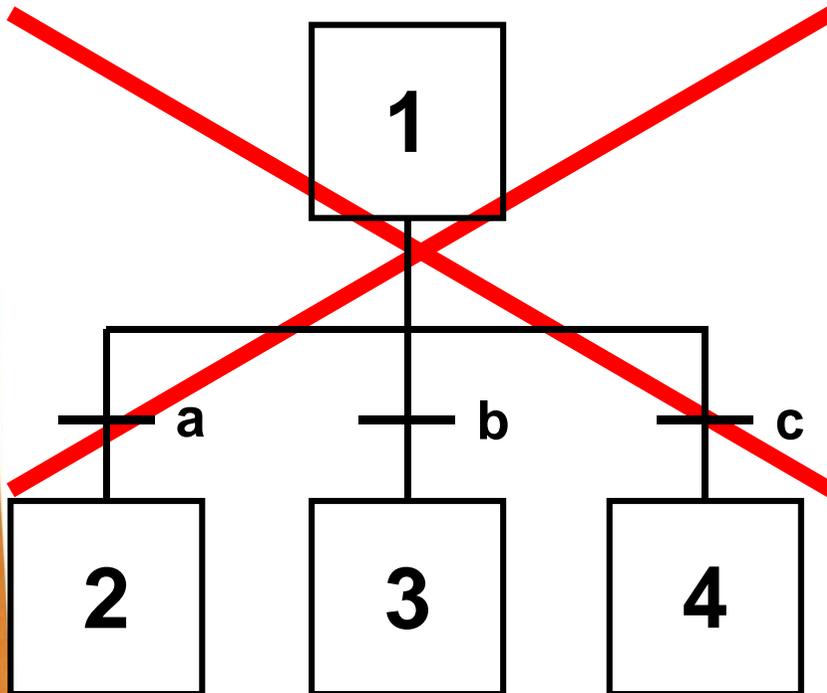
Les croisements de liaisons doivent être évités.



# VI. LIAISONS ORIENTÉES

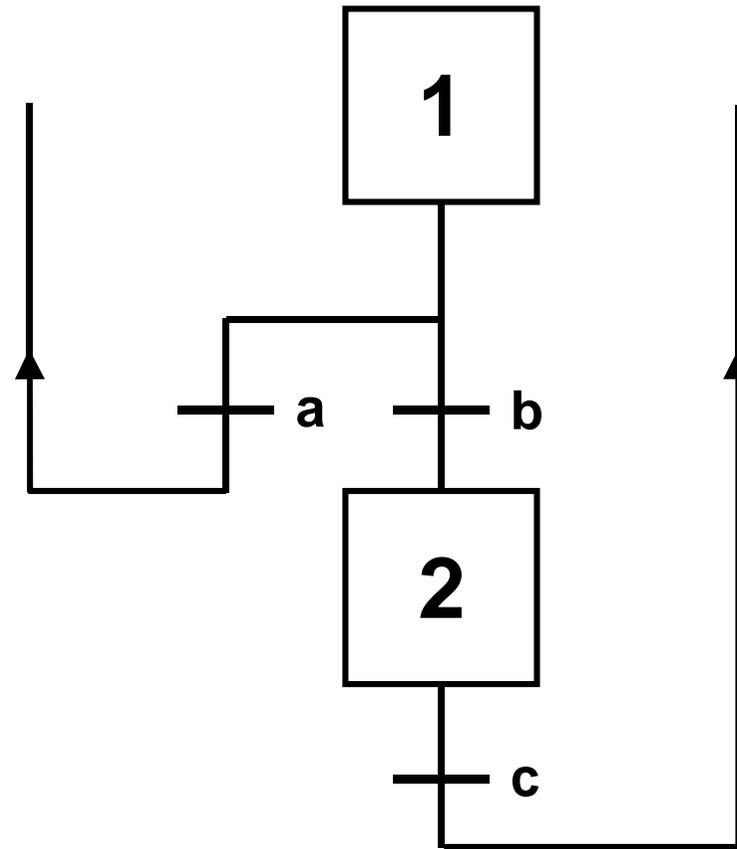
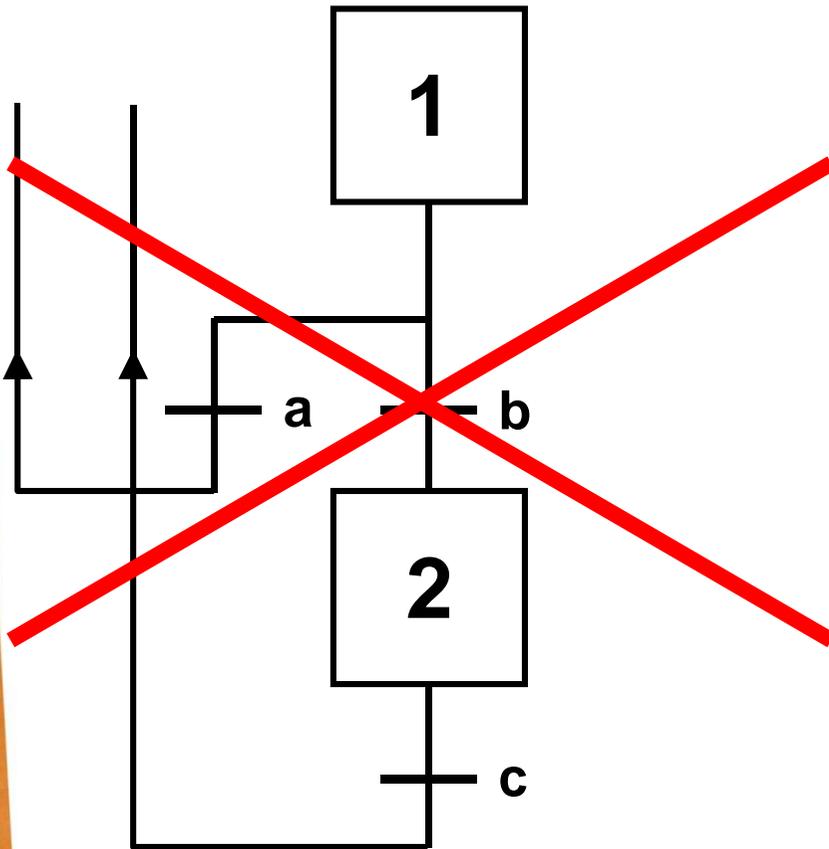
## 2. Règles de tracé

Les croisements de liaisons doivent être évités.



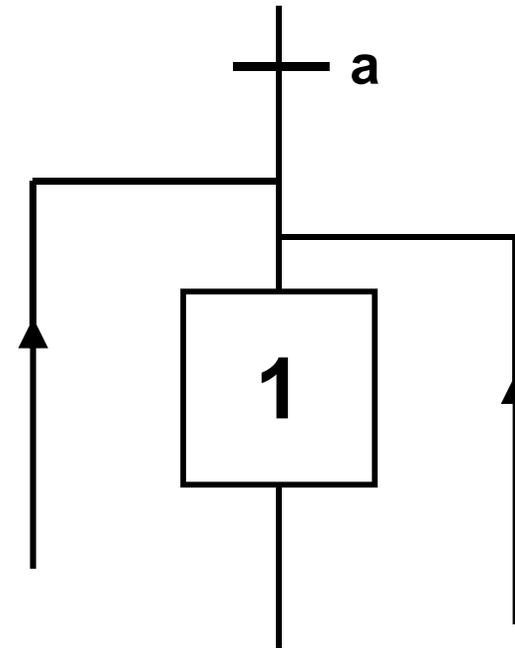
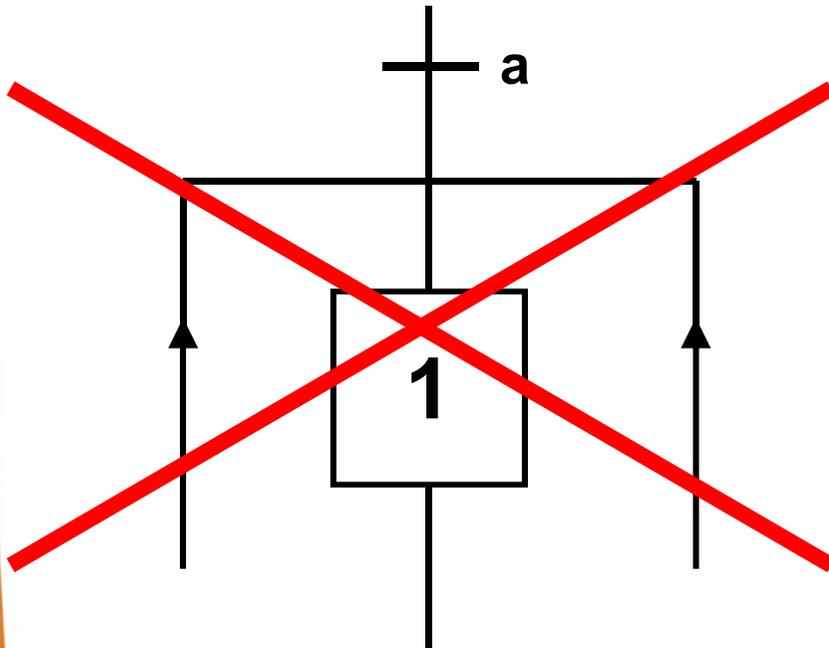
# VI. LIAISONS ORIENTÉES

## 2. Règles de tracé



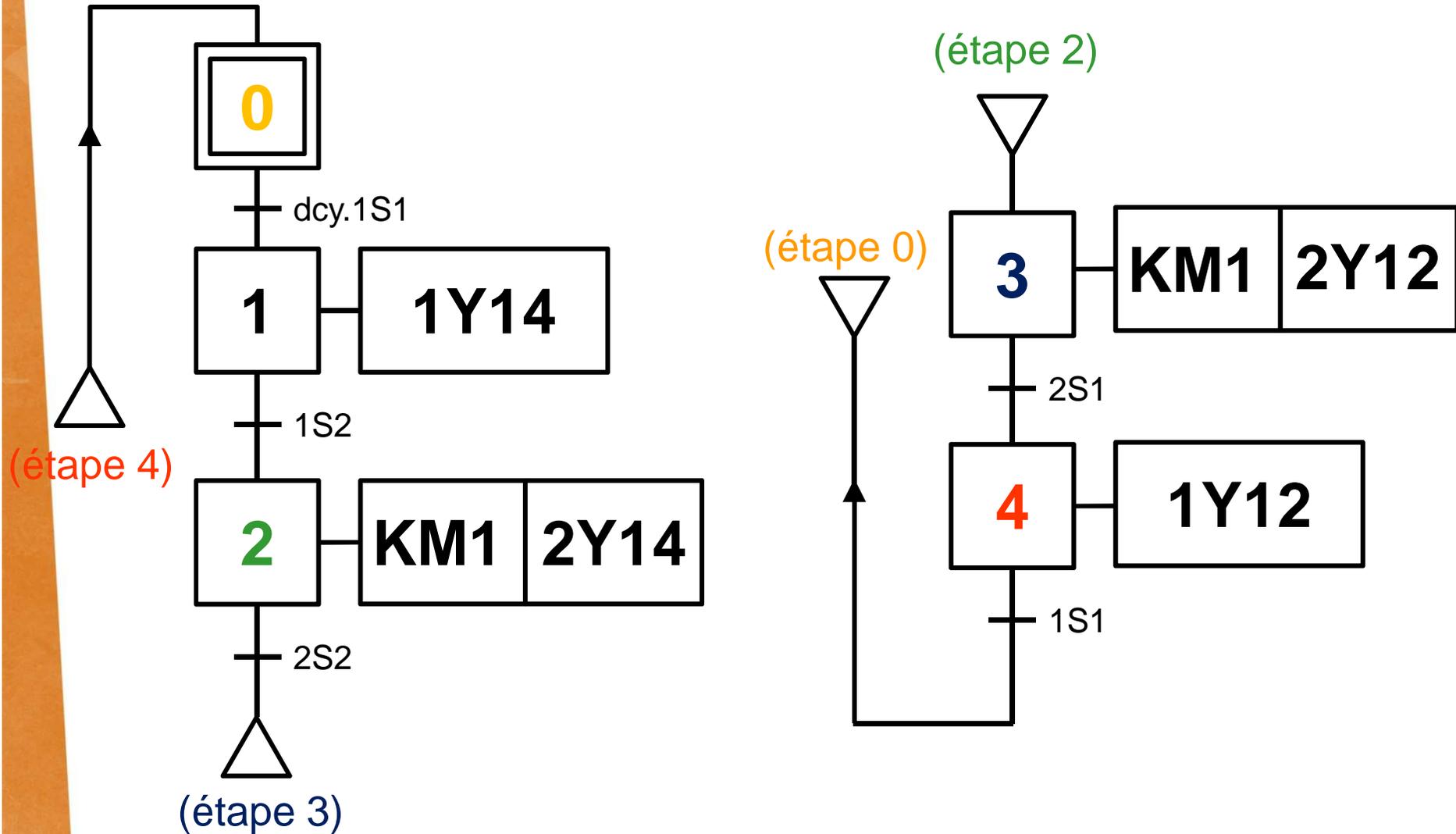
# VI. LIAISONS ORIENTÉES

## 2. Règles de tracé



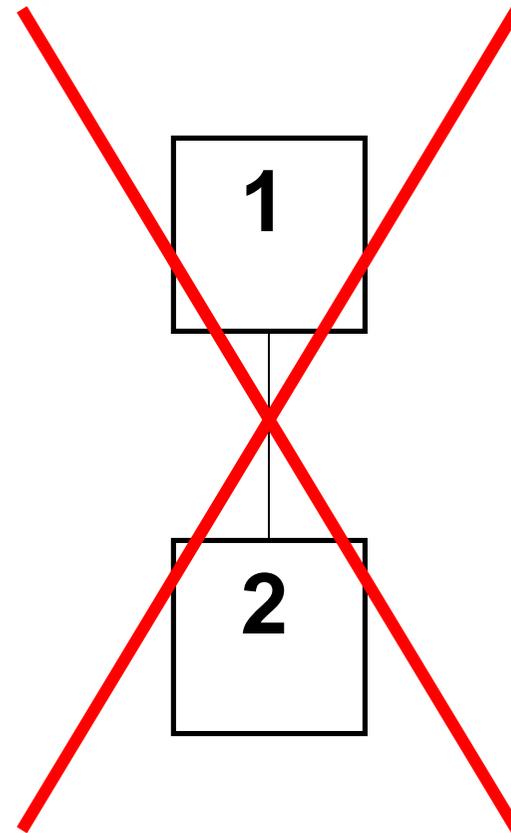
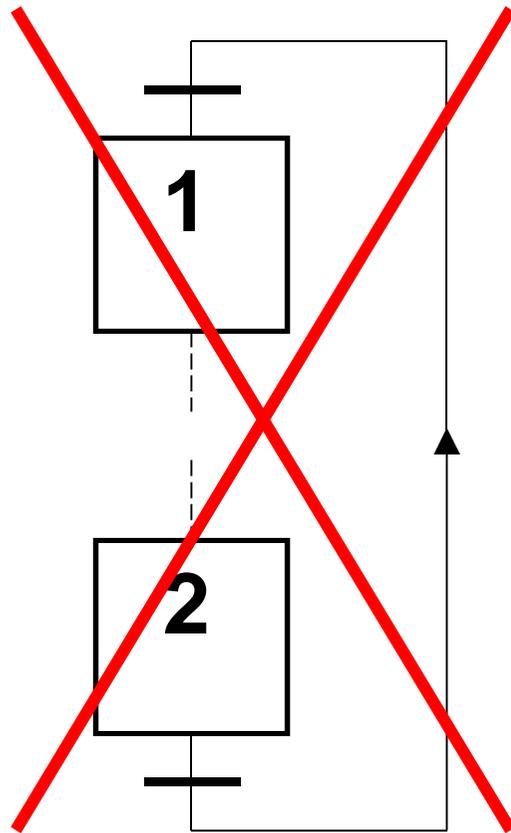
# VI. LIAISONS ORIENTÉES

## 3. Utilisation de renvois dans les liaisons



# VII. REGLE DE SYNTAXE

L'alternance **étape-transition** et **transition-étape** doit toujours être respectée quelle que soit la séquence parcourue.



# CONCLUSION

Ces règles de syntaxe permettent de construire un grafcet qui pourra être interprété sans aucune ambiguïté, ceci constitue une base indispensable pour **une communication technique efficace** entre les différents acteurs du développement d'un SA.

Elles sont nécessaires mais non suffisantes pour décrire le fonctionnement dynamique du GRAFCET, il manque **des règles d'évolution rigoureuses** permettant de décrire les changements d'activités des étapes en fonction des événements en entrée du système.