

## SYSTEMES SEQUENTIELS : LE GRAFCET

### I - Le modèle GRAFCET

#### 1. Introduction

C'est en 1975 qu'une idée apparaît, devant la complexité croissante des automatismes logiques, de créer un outil qui permette la représentation du cahier des charges d'un système, palliant ainsi les inconvénients des différentes méthodes existantes (principalement la lourdeur). Cette réflexion est conduite au sein de l'AFCET (**A**ssociation **F**rançaise pour la **C**ybernétique **E**conomique et **T**echnique) entre universitaires et industriels, et en 1977 un premier rapport sur l'outil **GRAFCET** paraît.

**GRAFCET** : **GRA**phe **F**onctionnel de **C**ommande **E**tape **T**ransition. L'écriture en majuscule GRAFCET correspond au modèle GRAFCET, et l'écriture en minuscule grafcet correspond au résultat graphique d'une étude du comportement d'un système.

Depuis le modèle GRAFCET a été normalisé (norme internationale), complété afin d'être implanté directement dans la partie commande d'un système (grafcet programmé), à l'aide de divers langages propres aux automates programmables.

Lors de la conduite d'un projet, différents grafkets seront élaborés, plus ou moins détaillés, en fonction de l'avancement de l'étude. Ils permettront la description selon différents points de vue, une vision globale du système automatisé, la description du fonctionnement du système d'un point de vue partie opérative, ou encore la description du fonctionnement attendu d'une partie commande identifiée.

#### 2. L'outil graphique

Le modèle GRAFCET : il s'agit d'un ensemble constitué d'éléments graphiques (nantis d'une syntaxe), d'une interprétation, et de règles d'évolution.

##### Les éléments graphiques :

**ETAPES**

**TRANSITIONS**

**LIAISONS ORIENTEES**

Le réseau alterné étape / transition forme l'ossature du grafcet.

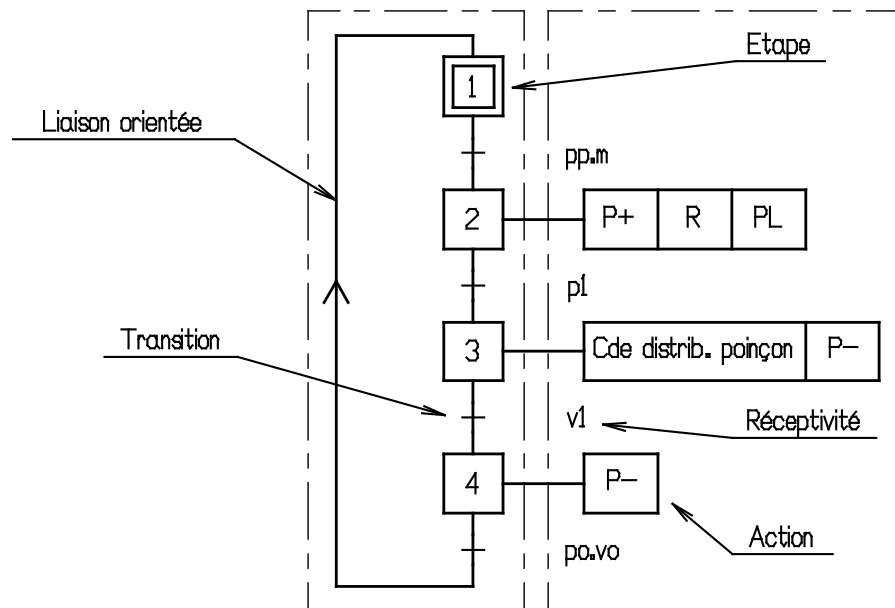
L'interprétation : elle traduit le comportement de la partie commande, vis à vis de ses entrées / sorties. Elle associe par exemple des expressions logiques aux étapes et aux transitions. Ce sont :

**ACTIONS**

**RECEPTIVITES**

Les règles d'évolution : au nombre de **cinq**, elles définissent le comportement dynamique de la partie commande.

On trouve alors la structure graphique suivante :



**ETAPE** : situation dans laquelle le comportement de la partie commande est invariant vis-à-vis de ses entrées et de ses sorties. Une étape est représentée par un carré, numéroté.

Une étape est **ACTIVE** ou **INACTIVE**. Un point à l'intérieur du carré est parfois utilisé pour l'étude du comportement dynamique du système, lorsque l'étape est active.

A une étape  $i$ , on peut associer une variable binaire  $x_i$  dont les états "0" et "1" sont associés respectivement à l'inactivité et à l'activité de l'étape  $i$ .

L'étape correspondant à l'initialisation du système est appelée **étape initiale**. Elle est représentée par un double carré. Il peut y avoir plusieurs étapes initiales dans un même grafcet.

**TRANSITION** : indique la possibilité d'évolution d'une situation à une autre situation. Le passage d'une situation à la suivante s'accomplit par le *franchissement* d'une transition, du haut vers le bas. L'évolution peut se faire entre deux ou plusieurs étapes. Une transition est représentée par une barre perpendiculaire à la liaison.

Pour faciliter la lecture, une transition peut être repérée par un identificateur, ou désignée par les étapes qu'elle sépare (l'identificateur sera placé à gauche).

**ACTION** : associée à une étape, une action n'est commandée que lorsque l'étape est active. On parle d'assignation sur état (en mode continu), ou d'affectation sur événement (en mode mémorisé) [cf. §IV].

**RECEPTIVITE** : équation booléenne logique associée à une transition. C'est une fonction logique des entrées, de variables auxiliaires et/ou de l'activité d'étapes. Elle permet de distinguer parmi toutes les variables du système, celles qui sont susceptibles de faire évoluer la partie commande par franchissement d'une transition.

**LIAISONS** : elles relient les étapes et les transitions, elles sont orientées. Le sens général est de haut en bas, s'il n'est pas indiqué. Des flèches doivent être utilisées dans le cas contraire, ou lorsque cela facilite la lecture. On peut utiliser parfois des lignes obliques, dans l'optique de clarifier le grafcet.

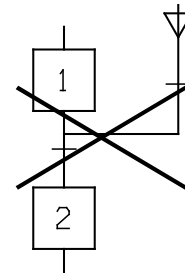
### 3. Règle de syntaxe

L'alternance **ETAPE - TRANSITION** et **TRANSITION - ETAPE** doit toujours être respectée quelle que soit la séquence parcourue :

# Deux étapes ne doivent jamais être reliées directement, elles doivent être séparées par une transition.

# Deux transitions ne doivent jamais être reliées directement, elles doivent être séparées par une étape.

La règle peut paraître évidente mais des erreurs sont souvent commises :



### 4. Les cinq règles d'évolution

#### REGLE N°1 : Situation initiale

La situation initiale du grafcet caractérise le comportement initial de la partie commande vis-à-vis de sa partie opérative. Elle correspond aux étapes actives au début du fonctionnement, soit à la mise en énergie de la partie commande.

#### REGLE N°2 : Evolution entre situations

Une transition est soit **VALIDEE**, soit **NON VALIDEE**. Elle sera validée lorsque toutes les actions immédiatement précédentes reliées à cette transition sont actives.

L'évolution de la situation du grafcet correspondant au **FRANCHISSEMENT** d'une transition ne peut se produire que lorsque :

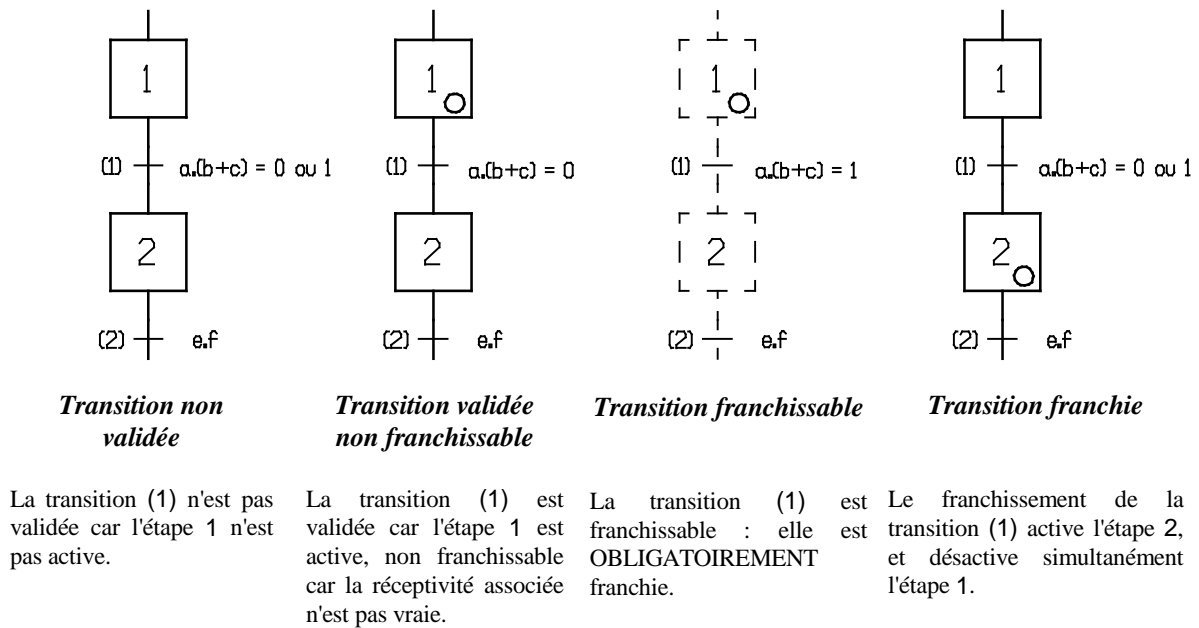
- la transition est **VALIDEE**, et
- la **RECEPTIVITE ASSOCIEE** à cette transition est **vraie**.

Lorsque ces deux conditions sont réunies, la transition devient **FRANCHISSABLE**, elle est alors **obligatoirement franchie**.

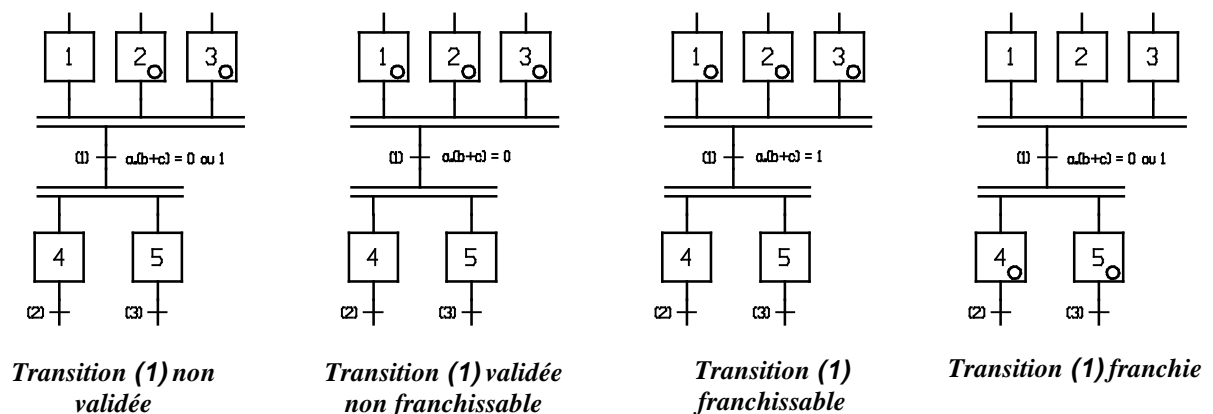
#### REGLE N°3 : Evolution des étapes actives

Le franchissement d'une transition entraîne **SIMULTANEMENT** l'activation de TOUTES les étapes immédiatement suivantes et la désactivation de TOUTES les étapes immédiatement précédentes.

Visualisation de l'évolution dynamique d'un grafcet, transition validée ou non, franchissable ou non, étape active ou non :



Autres cas d'évolution, activation de séquences parallèles et synchronisation de séquences (on retrouvera ces structures particulières, dans le paragraphe suivant).



### REGLE N°4 : Evolutions simultanées

Plusieurs transitions simultanément franchissables sont simultanément franchies.

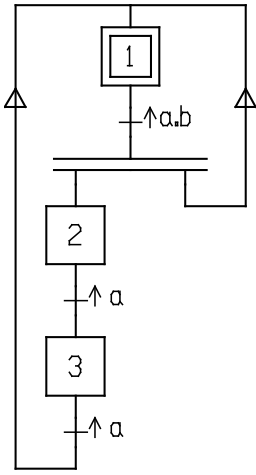
### REGLE N°5 : Activation et désactivation simultanée s

Si au cours du fonctionnement de l'automatisme , une même étape doit être **simultanément activée** et **désactivée**, elle reste **ACTIVE**.

Cette règle peut être illustrée par l'exemple ci-après, mais elle reste en pratique peu utilisée.

Lorsque la réceptivité ( $\uparrow a.b$ ) devient vraie, l'étape [1] étant active, la transition (1/2) est franchie. L'évolution implique l'activation de l'étape [2], et la désactivation de l'étape [1]. Mais dans le même temps, la liaison de droite impose l'activation de l'étape [1].

⇒ Celle-ci reste **ACTIVE**.



De même, si **b** reste vraie, un nouveau front montant de **a** entraîne l'évolution entre [2] et [3], rendant **ACTIVE** l'étape [3] tout en laissant **ACTIVE** l'étape [2]...

Pour visualiser l'évolution, il est possible de construire les chronogrammes...

## 5. Evolution fugace et non fugace

# **Evolution non fugace** : c'est le cas général, l'événement d'entrée ne provoque qu'un seul pas d'évolution (franchissement simultané d'une ou plusieurs transitions). L'état obtenu est **stable**.

# **Evolution fugace** : dans certain cas, l'application des règles d'évolution peut conduire à franchir successivement des transitions (en plusieurs pas d'évolution) si les réceptivités associées aux transitions postérieures sont déjà vraies lors du franchissement.

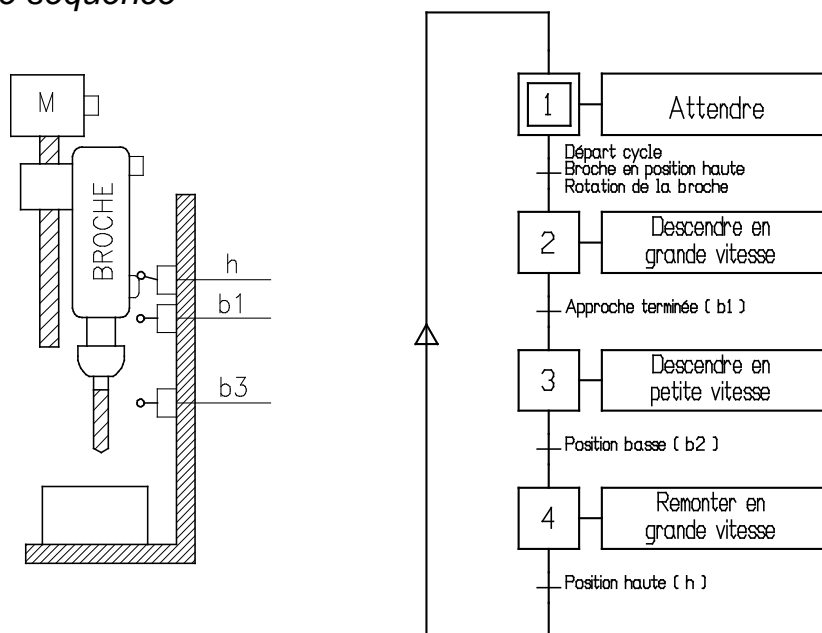
L'évolution correspondante, dite fugace, correspond à une succession d'états **instables**, durant lesquels les *étapes instables* ne sont pas activées. On dit qu'elles ont été virtuellement activées et désactivées, de même que les transitions associées ont été virtuellement franchies.

**Conséquence importante** : lors d'une évolution fugace, les actions associées aux étapes virtuellement activées ne sont pas commandées (**l'assignation sur état n'a pas lieu**). Seules les actions mémorisées seront prises en compte [cf. §IV].

## II – Structure générale d'un grafcet

### 1. Exemple de description à l'aide du modèle GRAFCET

*Cycle d'une seule séquence*



Fonctionnement de la perceuse :

La broche tourne en permanence.

La pièce est fixée par l'opérateur, et celui-ci donne l'information de départ du cycle.

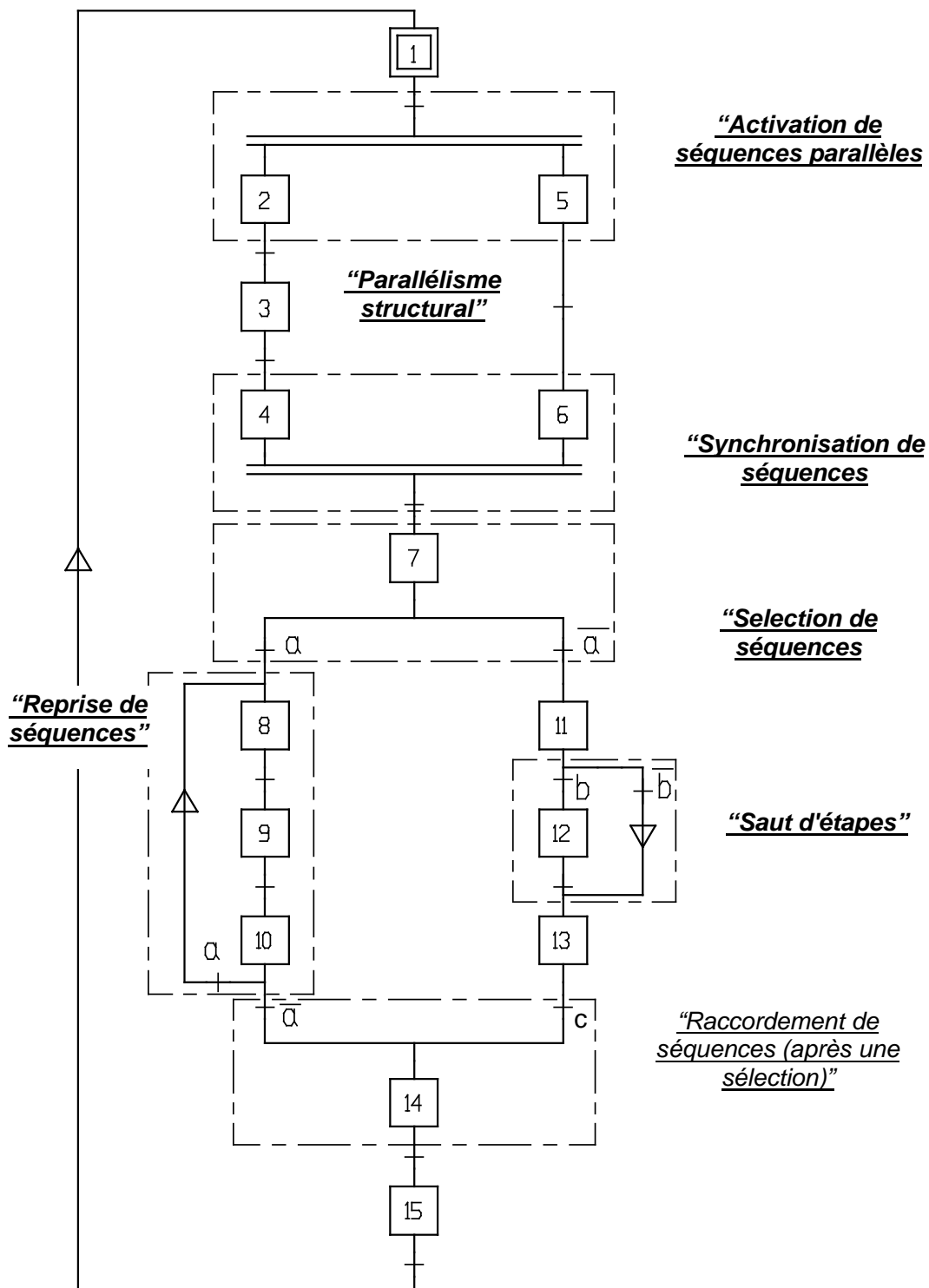
L'approche à grande vitesse (h, b1).

Le perçage en vitesse lente (b1, b3).

La remontée à grande vitesse.

**2. Les principales structures d'un grafcet**

Les principales structures d'un grafcet sont représentées ci-dessous. Pour chaque structure, il est impératif de respecter la règle de syntaxe et l'alternance étape / transition.



**Activation de séquences parallèles** : il s'agit d'une "distribution", le franchissement *d'une transition* entraîne l'activation des étapes suivantes.

**Sélection de séquences** : il s'agit d'une "sélection", seule une branche sera suivie. Il est préférable de rendre le OU exclusif (la structure ne l'impose pas), par la mise en place **de deux réceptivités** elles-mêmes exclusives.

**Synchronisation de séquences** : la transition n'est validée que lorsque toutes les étapes amont sont actives.

### III – Les différents points de vue d'une étude

#### 1. Points de vue et niveaux d'un grafcet

La description d'une partie commande d'un système automatisé doit intégrer plusieurs dimensions, son niveau de spécifications et le point de vue de la description. Il reste alors à définir le niveau de finesse de description : global, ou détaillé. Il y a deux principaux niveaux de spécifications : un premier niveau concernant les spécifications fonctionnelles, et un second niveau concernant les spécifications technologiques.

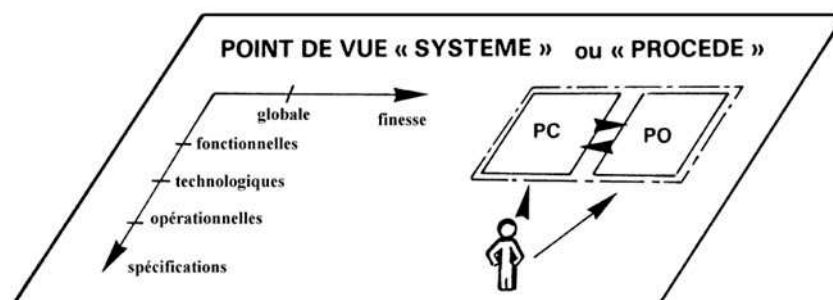
**Le premier niveau** permet de répondre à la question "*quelles sont les fonctions à réaliser dans une situation donnée?*" c'est une description en terme de fonction des comportements que doit avoir la partie commande face aux informations qu'elle reçoit. "**Fonctions globales à réaliser**".

**Le second niveau**, permet de savoir *comment chacune des fonctions est mise en œuvre dans le système*. C'est une description en terme de moyens, donc de solutions technologiques, du fonctionnement de la partie commande. "**Fonctions associées aux actionneurs**".

**Niveau de spécifications opérationnelles**, défini pour l'implantation du système automatisé dans le contexte de production (elles concernent les performances du système, la sûreté du fonctionnement, l'absence de pannes dangereuses...).

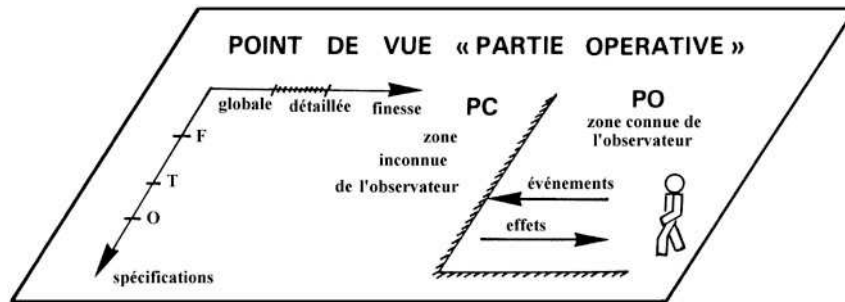
# Les points de vue sont au nombre de trois : le point de vue système, le point de vue partie opérative, et le point de vue partie commande. Ce sont les points de vue selon lequel l'observateur s'implique dans le fonctionnement du système, pour en donner une description.

**Le point de vue système ou procédé** : la description est faite par un observateur extérieur au système, sans distinguer la partie opérative et la partie commande (le système peut très bien ne pas avoir d'existence physique, au cours de sa conception).



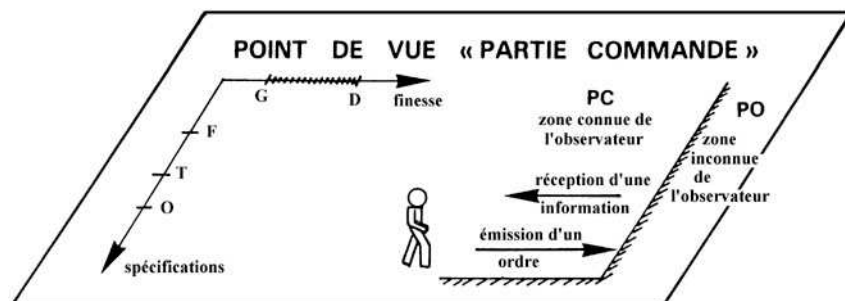
*Le grafcet décrit alors la coordination des tâches nécessaires à l'élaboration de la valeur ajoutée.*

**Le point de vue partie opérative** : l'observateur connaît la partie opérative, et ignore la partie commande.



*Le grafcet décrit le comportement attendu de la partie commande pour obtenir les effets souhaités sur la partie opérative. Evolution des actionneurs ou préactionneurs.*

**Le point de vue partie commande** : l'observateur connaît la partie commande, et ignore la partie opérative.

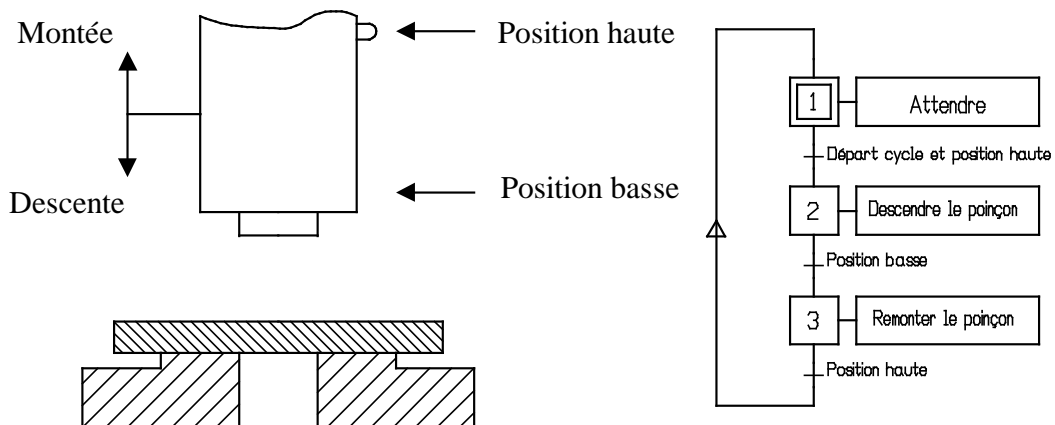


*Le grafcet décrit le comportement attendu de la partie commande, après avoir effectué les choix correspondants (évolution de l'automate).*

## 2. Exemple : Poinçonneuse semi-automatique

**2.1.** L'exemple de grafcet de la perceuse automatisée, correspond à une description fonctionnelle, il s'agit d'un grafcet point de vue système.

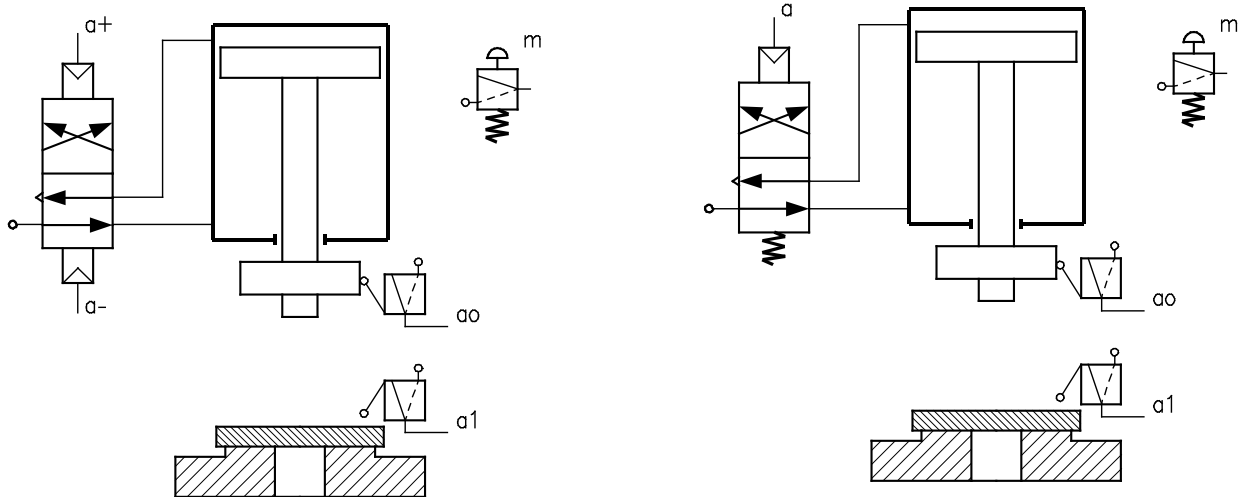
**2.2.** Le système présenté ci-dessous est une poinçonneuse semi-automatique, pour laquelle l'opérateur donne un ordre de départ, qui provoque la descente du poinçon, et sa remontée automatique. On peut facilement établir un grafcet point de vue système.



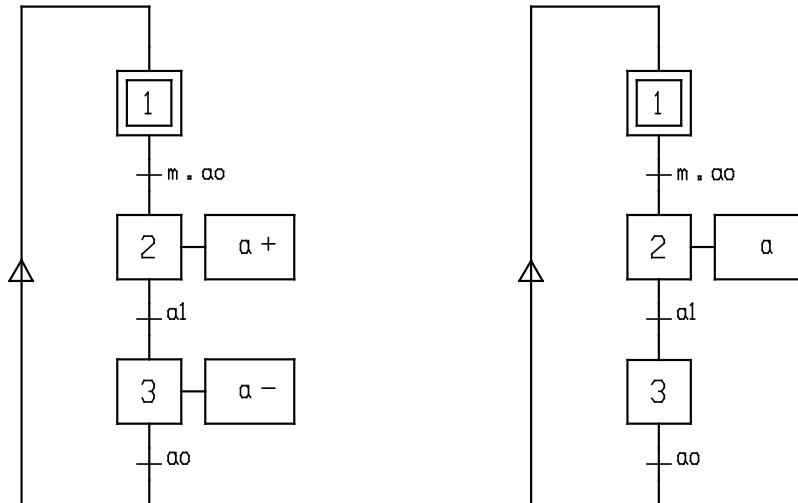


Si maintenant on effectue des choix technologiques pour la partie opérative, on peut élaborer le grafcet point de vue opérative.

Les deux figures ci-dessous précisent deux configurations matérielles différentes, de la partie opérative. On remarque dans cet exemple, que la frontière entre la partie commande, et la partie opérative est parfois ambiguë, ici les capteurs et préactionneurs sont inclus dans la partie opérative.



On établit alors deux grafkets différents, pour les deux configurations matérielles retenues.



### Remarque :

Le point de vue partie commande correspond aux ordres que doit donner l'automatisme (partie commande), en fonction des comptes rendus par la partie opérative. Il s'agit ici d'ordres du type "commander le distributeur a+" par exemple lorsqu'il faut faire descendre le poinçon (comptes rendus : marche et position haute).

Là encore selon l'avancement de la définition du système, cette description peut être plus ou moins détaillée.

## IV – Représentation graphique des éléments

Les éléments du GRAFCET possèdent une représentation symbolique normalisée. Ce paragraphe présente certains éléments importants du modèle GRAFCET, les éléments de bases étant déjà décrits dans les paragraphes précédents.

### 1. Variable temporisation

Il s'agit d'une variable logique qui peut être utilisée en réceptivité, mais aussi en condition d'assignation. Ces utilisations sont développées ci-après.

Représentation graphique générale :  $t_1/*/t_2$  avec  $t_1$  et  $t_2$  des durées, et  $*$  une variable logique. Caractéristique de la variable temporisation :

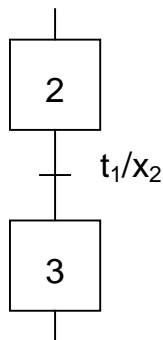
#  $t_1/*/t_2$  n'est égale à 1 qu'après un temps  $t_1$  depuis l'occurrence  $\uparrow*$ .

#  $t_1/*/t_2$  n'est égale à 0 qu'après un temps  $t_2$  depuis l'occurrence  $\downarrow*$ .

Utilisation usuelle :  $t_1/*$  ce qui correspond à  $t_2 = 0$ . La variable temporisation est remise à zéro dès le retour à zéro de la variable  $*$ .

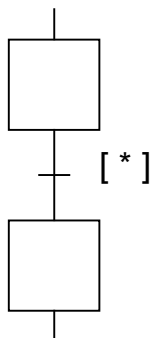
### 2. Réceptivités particulières

#### 2.1. Réceptivité dépendante du temps



L'étape 2 est temporisée  $t_1$  secondes par la variable temporisation.

#### 2.2. Valeur booléenne d'un prédicat



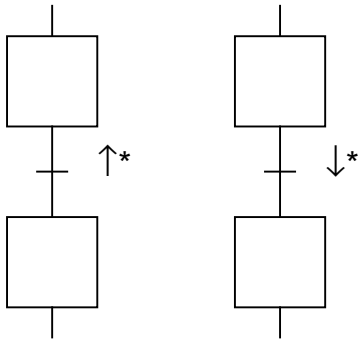
La notation  $[ * ]$  signifie que la variable vaut 1 lorsque le prédicat est vrai, et vaut 0 dans le cas contraire.

Exemples :

$[C_1 = 3]$

$[Température > 10^{\circ}]$

## 2.3. Fronts d'une variable



Les réceptivités  $\uparrow^*$  et  $\downarrow^*$  ne sont vraies que lorsque la variable change d'état, respectivement le front montant sera vrai lors du passage  $0 \rightarrow 1$  de la variable et le front descendant lors du passage  $1 \rightarrow 0$ .

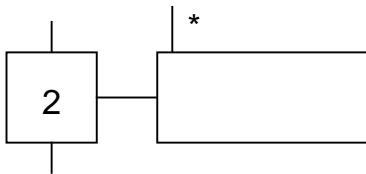
Remarque importante : l'état "vrai" d'un front ne possède pas de durée. C'est notamment utilisé dans les registres à décalage.

## 3. Actions continues (assignation sur état)

Il s'agit de l'association d'une action à une étape, qui permet d'indiquer qu'une variable de sortie a la valeur vraie si l'étape est active (et si la condition d'assignation éventuelle est vérifiée).

On appelle **assignation** le fait d'imposer une valeur (vraie ou fausse) à des variables de sortie.

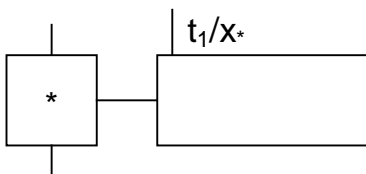
### 3.1. Condition d'assignation



\* est une proposition logique, appelée condition d'assignation.

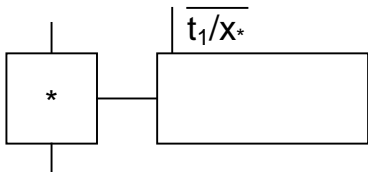
**Attention** : cela ne doit pas être un front de variable (voir actions mémorisées).

### 3.2. Action retardée – Action limitée



L'action associée ne sera commandée qu'après une durée  $t_1$  à partir de l'activation de l'étape.

**ACTION RETARDEE**



L'action associée sera limitée à une durée  $t_1$  à partir de l'activation de l'étape.

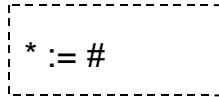
**ACTION LIMITEE**

## 4. Actions mémorisées (action sur événement)

C'est une action associée à un événement interne, qui permet d'indiquer qu'une variable de sortie prend une valeur et la garde (jusqu'à une modification ultérieure), lorsque l'événement se produit.

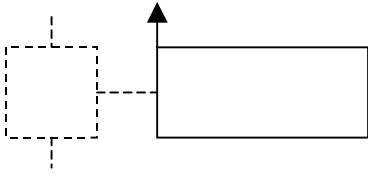
Une action mémorisée possède un libellé qui décrit comment la variable de sortie est affectée à une valeur déterminée selon la règle d'affectation. **Il doit être associé à un événement interne.**

#### 4.1. Libellé de l'action mémorisée

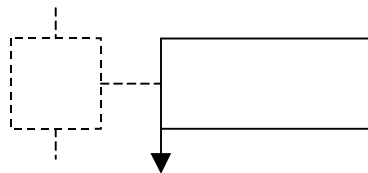


La valeur # est affectée à la variable \*, qui peut être booléenne ou numérique. Ainsi # peut être 1, 0, C+1...

#### 4.2. Action à l'activation - Action à la désactivation

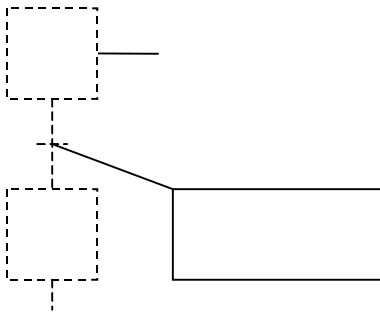


Une action à l'activation est une action mémorisée associée à l'ensemble des événements qui conduisent à l'activation de l'étape.



Une action à la désactivation est une action mémorisée associée à l'ensemble des événements qui conduisent à la désactivation de l'étape.

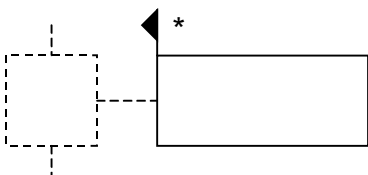
#### 4.3. Action au franchissement



Une action au franchissement est une action mémorisée associée à l'ensemble des événements qui conduisent au franchissement de la transition.

Cette représentation peut parfois être remplacée par une action à l'activation ou à la désactivation, mais pas systématiquement.

#### 4.4. Action sur événement



Une action sur événement est une action mémorisée associée à l'ensemble des **événements internes** décrits par l'expression logique \* est à l'activité de l'étape à laquelle elle est reliée.

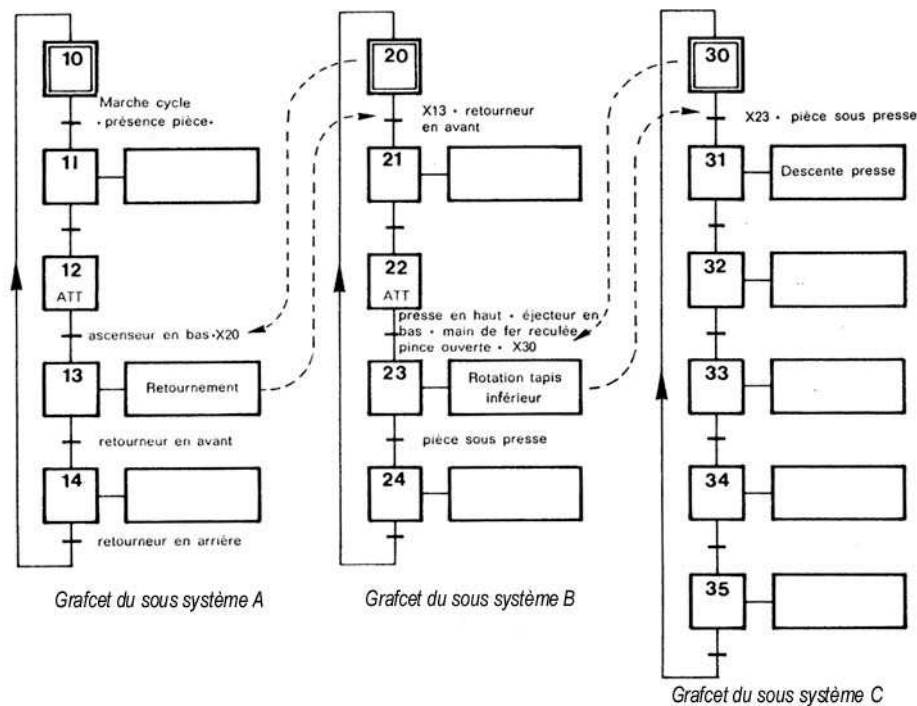
Cette expression logique doit **impérativement** contenir un ou plusieurs fronts de variables d'entrée.

**5. Commentaires dans un grafcet** : sur un grafcet un commentaire \* doit être indiqué entre guillemets : « \* »

### V – Simplification des représentations - autres structures

#### 1. Grafkets multiples

Dès lors que le système étudié comporte plusieurs sous-systèmes, il existe deux principales solutions de représentation : un grafcet unique comportant plusieurs étapes initiales ou des grafkets séparés, synchronisés par l'intermédiaire de variables d'activité d'étapes. Il est courant d'utiliser alors un grafcet de synchronisation, pour clarifier la lecture du fonctionnement.



*Grafkets multiples, avec visualisation des synchronismes*

La première représentation conduit à un grafcet plus complexe en général, mais qui a l'avantage de fournir une solution fiable en terme de synchronisation d'étapes. Elle peut être utilisée pour une partie opérative comportant deux ou trois sous systèmes.

La seconde représentation est plus simple à établir, mais plus délicate également. Des erreurs de synchronisme sont possibles, et la synchronisation entre étapes est beaucoup moins visible, ce qui est à l'encontre de l'objectif de lisibilité du grafcet.

Par contre cette représentation est très intéressante pour une implantation sur une machine, lorsque chaque sous système est géré par sa propre partie commande. C'est le mode de représentation des grafkets hiérarchisés, "arrêt d'urgence" ; "marche-arrêt"...

## 2. Les macro-étapes

La macro-étape est une représentation unique d'une partie détaillée de grafcet appelée *expansion* de la macro-étape. Elle ne possède pas toutes les propriétés d'une étape, car seule l'étape de sortie de son expansion valide ses transitions aval.

Son symbole est :



Il s'agit uniquement d'une représentation graphique qui a pour but de rendre plus lisible le grafcet, en représentant l'ensemble des étapes et transitions représentatives d'une partie du système à l'extérieur du grafcet. L'expansion agit alors comme un zoom.