

UNIVERSITE DE SOUSSE

Institut Supérieure des Sciences Appliquées et de Technologie de
Sousse



المعهد العالي للعلوم التطبيقية والتكنولوجيا بسوسة

Département de Génie Mécanique

Eléments de cours

Commande des systèmes industriels

(3^{ème} année formation d'Ingénieurs ITR)

Elaboré par :

- *M^r. HOUIDI Ajmi*

Année Universitaire 2010/2011

ISSAT SOUSSE

Cité Ettafela Ibn Khaldoun, 4003, Sousse-Tunisie

Tél : 216-73.382.656

e-mail : issats@issatso.rnu.tn

Télécopie : 216-73.382.658

Généralités

Chaque processus industriel de fabrication ou de transformation se compose d'un ensemble de machines destinées à réaliser la fabrication ou la transformation considérée.

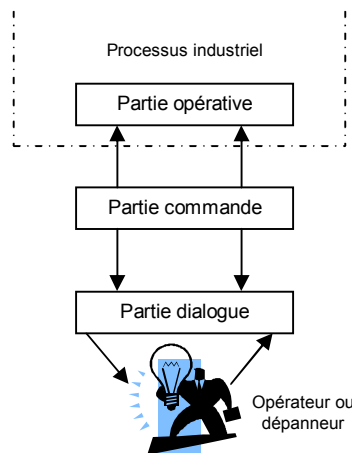
Chaque machine ou partie opérative comprend un ensemble de moteurs, vérins, vannes et autres dispositifs qui lui permet de fonctionner.

Ces moteurs, vérins, vannes et autres dispositifs s'appellent actionneurs. Ils sont pilotés par un automate ou partie commande (PC).

Cette partie commande élabore les ordres transmis aux actionneurs à partir des informations fournies par la machine au moyen d'interrupteurs de position, thermostats, manostats et autres dispositifs appelés capteurs.

La partie commande reçoit également des informations transmises par un opérateur en fonctionnement normal, ou un dépanneur en cas de réglage ou de mauvais fonctionnement de la partie commande ou de la partie opérative.

Entre la partie commande et l'homme se trouve la partie dialogue qui permet à ce dernier de transmettre des informations au moyen de dispositifs adaptés (boutons poussoirs, commutateurs, etc.)



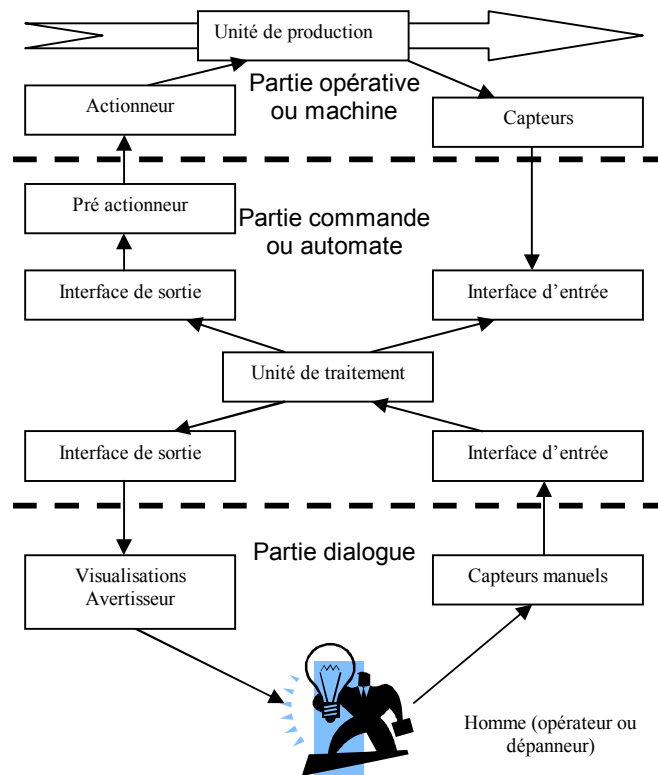
Structure générale d'un système automatisé

De même, la partie commande retourne vers l'homme des informations sous des formes compréhensibles par lui (voyant, afficheurs, cadrans, etc.).

Ainsi, entre l'homme et la partie opérative, s'instaure un dialogue homme – machine dont l'importance naguère sous-estimée est aujourd'hui reconnue et qui est actuellement l'objet de nombreuses études.

2 Structure des automatismes

Partant des définitions de base énoncées au paragraphe précédent, nous allons maintenant détailler les éléments constitutifs d'un automatisme (figure suivante).



Analyse de la partie opérative :

La partie opérative se compose de trois ensembles :

- L'unité de production dont la fonction est de réaliser la fabrication ou la transformation pour laquelle elle remplit un rôle dans le processus industriel,
- Les actionneurs qui apportent à l'unité de production l'énergie nécessaire à son fonctionnement à partir d'une source d'énergie extérieure (cas d'un moteur, par exemple). Ces actionneurs peuvent aussi prélever de l'énergie sur l'unité de production pour la retourner vers un récepteur d'énergie extérieur (cas d'un frein, par exemple).
- Les capteurs qui créent, à partir d'informations de natures divers (déplacement, température, etc.), des informations utilisables par la partie commande (ouverture ou fermeture d'un circuit électrique, par exemple)

Analyse de la partie commande

La partie commande se compose de quatre ensembles :

- Les interfaces d'entrée qui transforment les informations issues des capteurs placés sur la partie opérative ou dans la partie dialogue en informations de nature et d'amplitude compatible avec les caractéristiques technologiques de l'automate.
- Les interfaces de sortie qui transforment les informations élaborées par l'unité de traitement en informations de nature et d'amplitude compatibles avec les caractéristiques technologiques des préactionneurs d'une part, des visualisations et avertisseurs d'autre part ;
- Les préactionneurs qui sont directement dépendants des actionneurs et sont nécessaires à leur fonctionnement (démarrreur pour un moteur, distributeur pour un vérin, etc) ;
- L'unité de traitement qui élabore les ordres destinés aux actionneurs en fonction des informations reçues des différents capteurs et du fonctionnement à réaliser.

Analyse de la partie dialogue

La partie dialogue se compose de deux ensembles :

- Les visualisations et avertisseurs qui transforment les informations fournies par l'automate en informations perceptibles par l'homme (informations optiques ou sonores) ;
- Les capteurs qui transforment les informations fournies par l'homme (action manuelle sur un bouton-poussoir, par exemple) et informations exploitables par l'automate.

LE GRAFCET

1. Introduction :

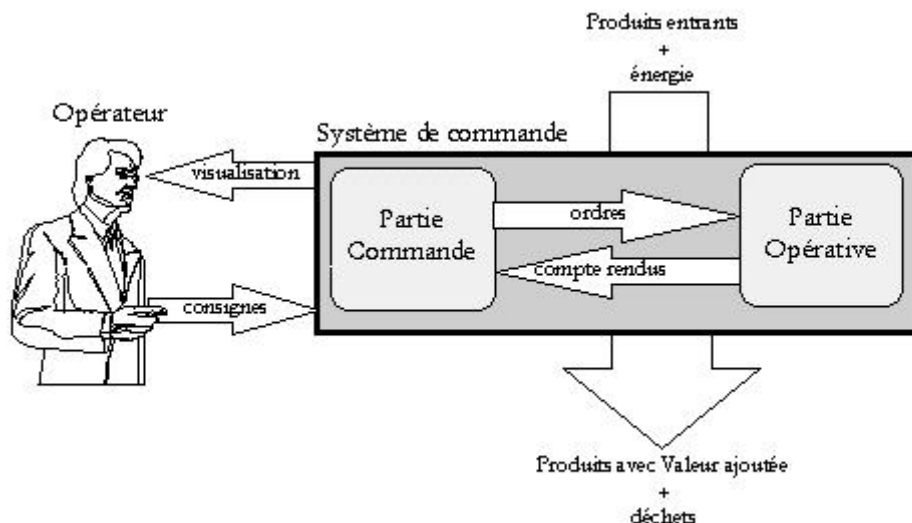
Le GRAFCET est un diagramme fonctionnel dont le but est de décrire graphiquement les différents comportements d'un automate séquentiel.

Créé par l'AFCEC (Association Française pour la Cybernétique Économique et Technique), le Grafcet est la synthèse d'une vingtaine de systèmes de description proposés à l'origine (1976).

Sa promotion en a été faite par l'ADEPA (Agence pour le Développement de la Production Automatisée) puis a été acceptée par les instances internationales de normalisation, notamment par le Comité Électrotechnique International dans sa publication 848 de l'année 1988 d'où proviendront de large extraits de ce cours (CEI 848)

Domaine d'application du GRAFCET

Le GRAFCET (Graphe Fonctionnel de Commande Etape Transition), également appelé Diagramme Fonctionnel en Séquence ou Sequential Function Chart (SFC), permet "l'établissement des descriptions de la fonction et du comportement des systèmes de commandes en établissant une représentation graphique indépendante de la réalisation technologique".



La figure ci-dessus montre la structure d'un système de commande (ou système automatisé de production) ainsi que ses relations avec l'opérateur et avec les produits, objets de la production. Le système de commande se décompose en une partie opérative (PO) et une partie commande (PC). La partie opérative est composée du processus physique que l'on souhaite piloter (elle comprend notamment les actionneurs, pré-actionneurs et capteurs). La partie commande est constituée de l'automatisme qui élabore les ordres destinés au processus et les sorties externes (visualisation) à partir des comptes rendus de la partie opérative, des entrées externes (consignes) et de l'état du système.

Plus pragmatiquement, le GRAFCET est destiné à représenter des automatismes logiques séquentiels, c'est à dire des systèmes événementiels dans lesquels les informations sont de type booléennes (tout ou rien) ou peuvent s'y ramener (numériques). Le GRAFCET est utilisé généralement pour spécifier et concevoir le comportement souhaité de la partie commande d'un système de commande mais il peut également être utilisé pour spécifier le comportement attendu de la partie opérative ou bien de tout le système de commande.

Destiné à être un moyen de communication entre l'automaticien et son client, le GRAFCET est un outil utilisé pour la rédaction du cahier des charges d'un automate. Cependant un des points forts du GRAFCET est la facilité de passer du modèle à l'implantation technologique de celui-ci dans un automate programmable industriel. Le GRAFCET passe alors du langage de spécification au langage d'implémentation utilisé pour la réalisation de l'automatisme. On parle ainsi de grafcet de spécification et de grafcet de réalisation. Les chapitres suivants seront donc consacrés à la définition du GRAFCET et à son utilisation en tant que langage d'implémentation.

2. Exemple Poinçonneuse semi-automatique :

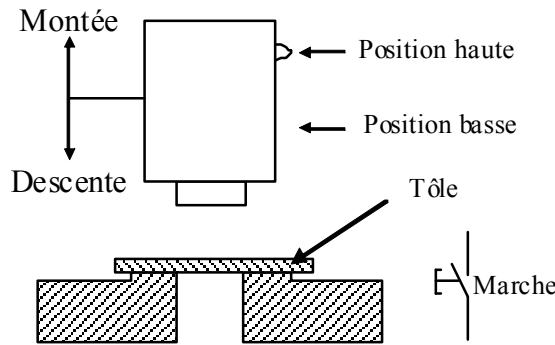


Figure 1 Constitution de la poinçonneuse

La poinçonneuse représentée très schématiquement ci-contre se compose d'une table fixe recevant la tôle à poinçonner et d'un poinçon mobile. Considérons la poinçonneuse en sa position origine au repos, poinçon en haut (configuration initiale).

L'opérateur en donnant l'information "Marche" provoque automatiquement la descente du poinçon puis sa remontée en position repos. Nous dirons alors que la poinçonneuse décrit un cycle de fonctionnement.

Une telle machine présente successivement trois comportements différents : repos, descente et montée. Nous appellerons étape chacun de ces comportements.

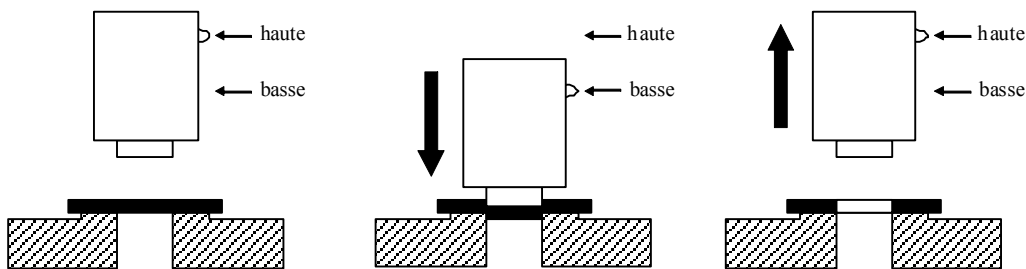


Figure 2 Description des étapes

Ces trois étapes sont :

- ETAPE 1 - Comportement : La poinçonneuse est au repos
- ETAPE 2 - Comportement : Descendre le poinçon
- ETAPE 3 - Comportement : Remonter le poinçon

Il s'agit maintenant de préciser ce qui provoque un changement de comportement de la machine c'est à dire les conditions logiques qui déterminent le passage d'un comportement à un autre. Nous qualifierons chaque passage d'un comportement à un autre comme étant le franchissement d'une transition pour bien montrer son irréversibilité.

Par exemple, le passage de la position de repos (étape 1) à la descente du poinçon (étape 2) ne peut s'effectuer que si l'opérateur fournit l'information "Marche" et que si le poinçon est en position haute (Conditions Initiales (CI)).

Ces deux informations, "Marche" et "Conditions Initiales", constituent la condition de transition appelée "**Réceptivité**" associée à la transition de l'étape 1 vers l'étape 2 ($t_{1 \rightarrow 2}$).

Cette description de fonctionnement de la poinçonneuse est représentée par le GRAFCET suivant :

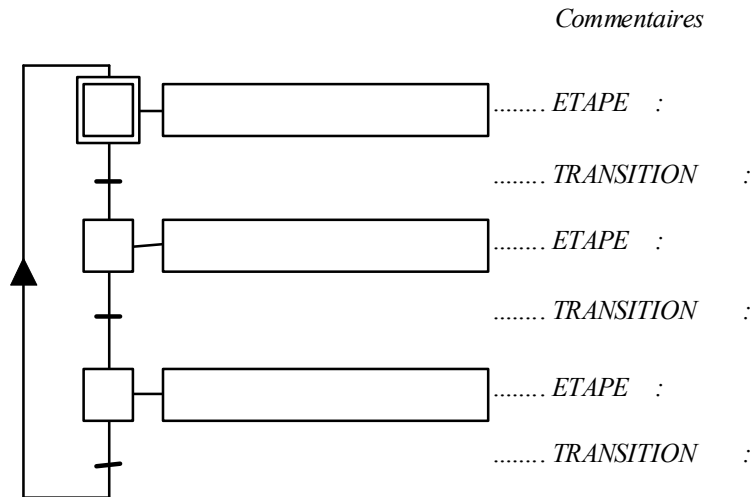


Figure 3 *Grafcet fonctionnel de la poinçonneuse*

Nous remarquerons que ce Grafcet correspond à une succession alternée d'étapes et de transitions.

Nous associerons :

- À chaque étape, le comportement ou l'action(s) à obtenir.
- À chaque transition, les informations permettant leur franchissement sous la forme d'une condition logique ou réceptivité.

Nous pouvons donc, dans un premier temps, définir une étape comme une situation du cycle de fonctionnement pendant laquelle le comportement de l'automatisme demeure constant.

Sous une autre forme, le changement de comportement provoque obligatoirement le passage à une autre étape.

Sur la machine le comportement de l'automatisme se manifestera par des actions ou plus exactement par des ordres envoyés vers les organes chargés d'exécuter ces actions.

Sur la poinçonneuse deux actions sont effectuées :

- La descente du poinçon associée à l'étape 2 ;
- La remontée du poinçon associée à l'étape 3.

Une étape est soit active, soit inactive, et les actions associées à une étape sont effectives que lorsque celle-ci est active.

Les transitions indiquent, avec les liaisons orientées, les possibilités d'évolution entre étapes.

La condition de transition est appelée réceptivité, car elle permet de distinguer, parmi toutes les informations disponibles, uniquement celles qui, à un instant donné, sont susceptibles de provoquer un changement de comportement. Soit encore que dans une étape, l'automatisme n'est réceptif qu'à ces informations.

La symbolisation utilisée dans l'écriture des Grafcet peut être résumée par la figure suivante :

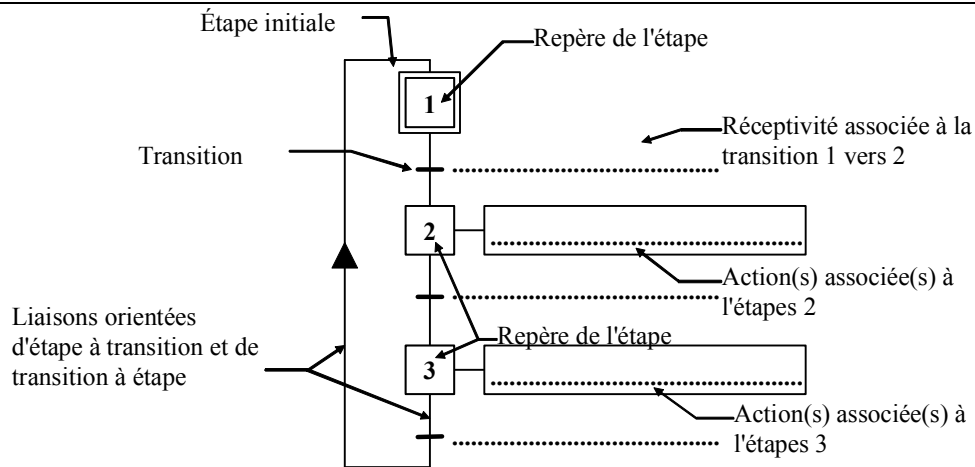


Fig. 4. Symbolisation du grafcet

Chaque étape est représentée par un carré repéré numériquement. En addition à ce repère un nom symbolique peut être ajouté représentatif de la fonction principale de l'étape (ex. : attente, fin, etc.).

Les étapes initiales, représentant les étapes actives au début du fonctionnement, se différencient en doublant les côtés du carré. Les numéros sont utilisés une seul fois.

Les actions associées sont décrites de façon littérale ou symbolique à l'intérieur d'un ou plusieurs rectangles de dimension quelconque reliés à la partie droite de l'étape.

Les transitions sont repérées par des barres horizontales.

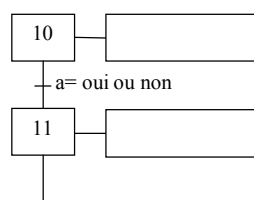
Les réceptivités sont inscrites, sauf cas particulier, à droite de chaque transition.

2.1 Structure de base :

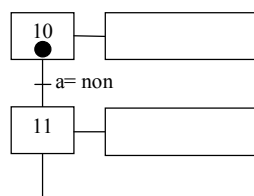
2.2 Règles d'évolution du grafcet

En considérant que l'étape active est celle dans laquelle un point est dessiné, les situations suivantes sont possibles

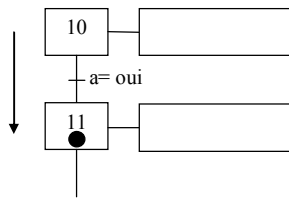
- Aucune étape n'est active. Il ne se passe rien quelque soit l'état de la réceptivité



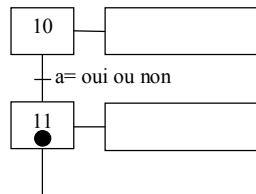
- L'étape 10 est active, la transition est valide mais les conditions représentées par la réceptivité ne sont pas remplies. Les actions liées à l'étape 10 sont exécutées. Le point représenté au niveau de l'étape 10 est une schématisation de l'activation de cette étape.



- L'étape 10 est active, les conditions représentées par la réceptivité sont remplies, la transition est franchie. Le passage à la situation suivante est immédiat.



- L'étape 11 est active. Quelque soit l'état de la réceptivité (a) associée à la transition $t_{10 \rightarrow 11}$ ce sont les actions liées à l'étape 11 qui sont maintenant exécutées. On dit que le grafset n'est pas réceptif à la réceptivité (a).



Les évolutions courantes peuvent être représentées par les structures de base suivantes.

2.3 Séquence unique :

Une séquence unique est composée d'une suite d'étapes qui seront activées les unes après les autres. Dans cette structure, chaque étape est suivie par une seule transition et chaque transition par une seule étape.



Figure 4 Exemple d'une séquence unique

La figure 4 illustre l'exemple d'une séquence unique. Une évolution de l'étape 3 vers l'étape 4 pourra se produire uniquement si l'étape 3 est active ($X3 = 1$) et la condition de transition "c" associée à la transition $t_{3 \rightarrow 4}$ est vraie ($c = 1$). Il en est de même lorsque l'étape 4 est active ($X4 = 1$). Si la réceptivité "d" est vraie ($d = 1$), le franchissement de la transition $t_{4 \rightarrow 5}$ conduit vers l'étape 5 et désactive l'étape 4.

2.4 Sélection de séquence (mode de sélection) :

2.4.1 Début de sélection de séquence (divergence de sélection de séquence) :

Une sélection entre une ou plusieurs séquences est représentée (sous la ligne horizontale) par autant de symboles de transition qu'il y a d'évolutions différentes. Aucun symbole commun de transition n'est permis au-dessus de la ligne horizontale.

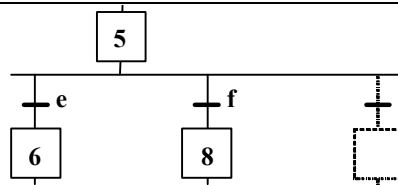


Figure 5 Exemple d'une sélection de séquence

La figure 5 présente un exemple de sélection de séquence. Une évolution s'effectuera de l'étape 5 vers l'étape 6 si l'étape 5 est active ($X_5 = 1$) **et** si la condition de transition "e" est vraie ($e = 1$) ou de l'étape 5 vers l'étape 8 si la condition de transition "f" est vraie ($f = 1$).

Note : de façon à sélectionner une seule séquence, il est nécessaire que les conditions de transition associées à ces séquences soient exclusives afin de ne pas être vraies en même temps. Il est toujours possible d'attribuer un ordre de priorité dans l'annotation des conditions de transitions (voir fig. 6).

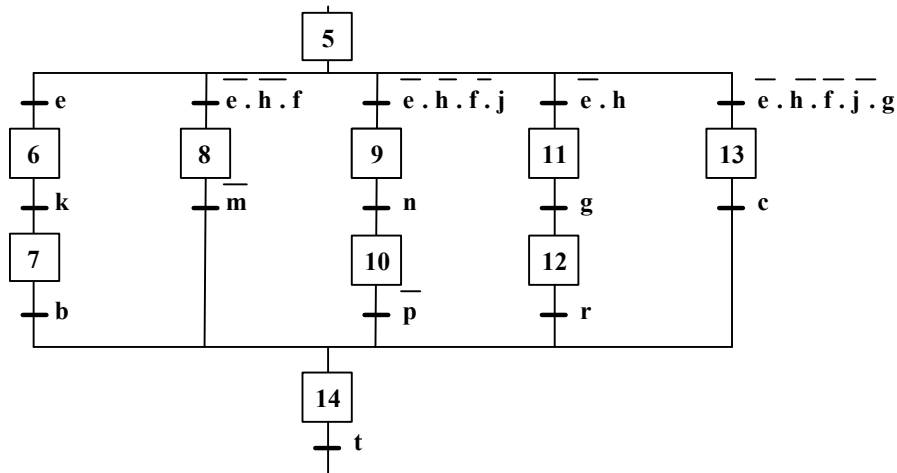
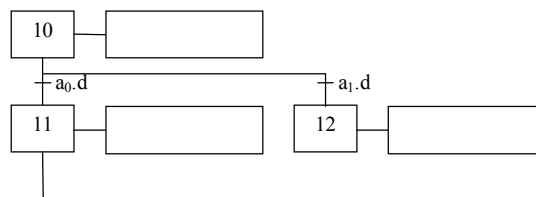


Figure 6 Exemple d'une sélection de séquence après l'étape 5, spécifiée par annotation des conditions de transitions

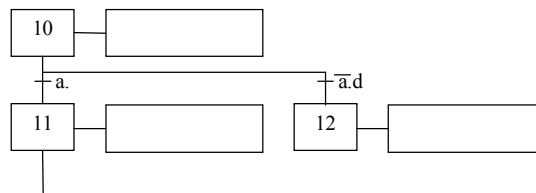
Exclusivité technologique

L'exclusivité peut être assurée technologiquement comme le montre la figure suivante :



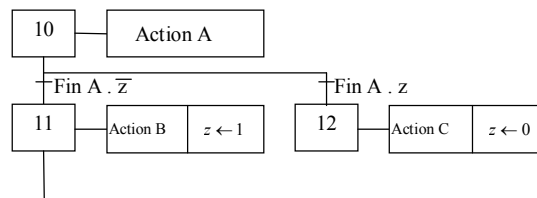
Les réceptivités a_0d et a_1d sont technologiquement exclusives par l'intermédiaire des capteurs a_0 et a_1 qui sont placés de sorte qu'ils ne soient pas actionnés en même temps (simultanément).

Exclusivité logique avec priorité



Les réceptivités a et $\bar{a}d$ sont logiquement exclusives avec priorité à la transition $t_{10 \rightarrow 11}$ si a et d sont mis à 1 en même temps.

Sélection alternative par bit de drapeau



Le bit z associé à l'étape 11 et à l'étape 12 joue le rôle de drapeau (flag) ou plus simplement l'aiguilleur automatique. Si on se trouve dans la situation $s=\{10\}$ avec la réceptivité (Fin A et $z=0$) on évolue vers la situation $s=\{11\}$. L'action d'affectation $z \leftarrow 1$ associée à l'étape 11 met la variable z à la valeur 1. Au prochain passage par la situation $s=\{10\}$, comme $z=1$, le grafcet évolue vers la situation $s=\{12\}$. L'action d'affectation $z \leftarrow 0$ associée à l'étape 12 remet la variable z à 0. Ainsi à chaque passage dans la situation $s=\{10\}$, on se trouve orienté alternativement, soit vers l'étape 11 ou vers l'étape 12.

2.4.2 Fin de sélection de séquence (convergence de sélection de séquence) :

La convergence de plusieurs étapes vers une séquence commune se représente (au-dessus de la ligne horizontale), par autant de symboles de transition qu'il y a de séquences à regrouper. Aucun symbole commun de transition n'est permis en dessous de la ligne horizontale.

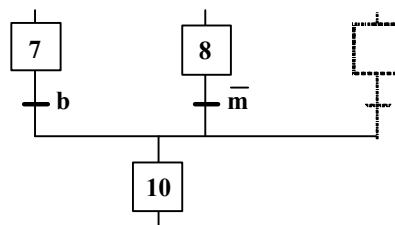


Figure 7 Exemple de fin de sélection de séquences

La figure 7 représente un exemple de fin de séquence. Une évolution pourra se produire de l'étape 7 vers l'étape 10 si l'étape 7 est active et si la condition de transition "b" est vraie ($b = 1$), ou de l'étape 8 à l'étape 10 si l'étape 8 est active ($X8 = 1$) et si la condition de transition " \bar{m} " est vraie ($\bar{m} = 1$).

2.5 Séquences simultanées (parallélisme structural) :

Début de séquences simultanées (divergence de séquences simultanées) :

Lorsque le franchissement d'une transition conduit à activer plusieurs séquences en même temps, ces séquences sont dites simultanées. Après leur activation simultanée, l'évolution des étapes actives dans chacune de ces séquences devient alors indépendante. Un seul symbole commun de transition est permis au-dessus de la double ligne horizontale de synchronisation (ISO 5807, symbole n° 9.2.2.5).

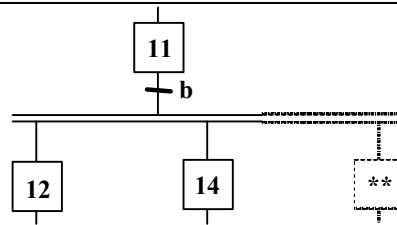


Figure 8 Exemple de début de séquences simultanées

La figure 8 illustre l'exemple de séquences simultanées. Une évolution de l'étape 11 ne pourra s'effectuer vers les étapes 12, 14, et ** que si l'étape 11 est active ($X_{11} = 1$) et si la condition de transition "b" associée à la transition commune ($t_{11 \rightarrow 12,14,**}$) est vraie ($b = 1$).

2.5.1 Fin de séquences simultanées (convergence de séquences simultanées) :

Pour synchroniser les convergences de plusieurs séquences en même temps, la structure suivante est utilisée.

Un seul symbole commun de transition peut être placé sous la double ligne horizontale de synchronisation. La divergence et la convergence de séquences simultanées peuvent être effectuées en une ou plusieurs fois (voir figure 10).

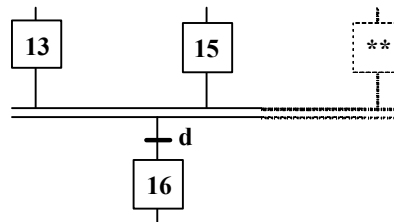


Figure 9 Exemple de fin de séquences simultanées

La figure 9 illustre l'exemple de convergence de séquences simultanées. Une évolution des étapes 13, 15 et ** ne pourra s'effectuer vers l'étape 16 que si les étapes immédiatement au-dessus de la double ligne sont actives et si la condition de transition "d" associée à la transition commune est vraie ($d = 1$).

Remarque : A la fin de chaque séquence impliquée dans un parallélisme structural il faut toujours laisser une étape d'attente afin de ne pas solliciter le système dans le cas où il y a un problème au niveau d'une séquence.

Sélection par une étape de synchronisation

Les étapes 16 et 18 autorisent respectivement le franchissement des transitions $t_{14,16} \rightarrow 15$ et $t_{18,20} \rightarrow 17$. Les étapes 16 et 18 sont dites étapes de synchronisation. L'étape 14 est une étape d'attente.

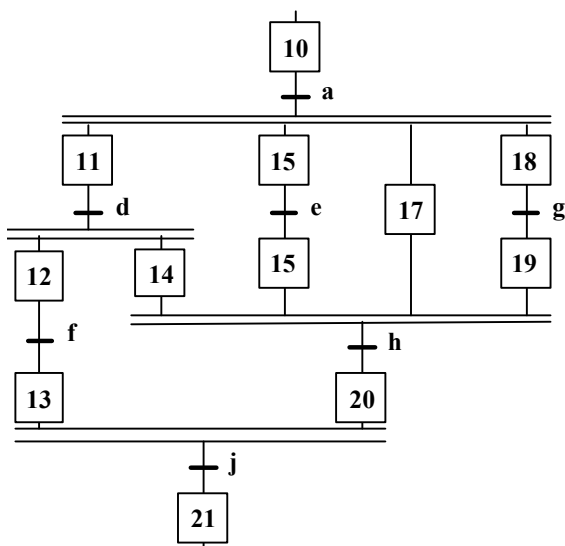
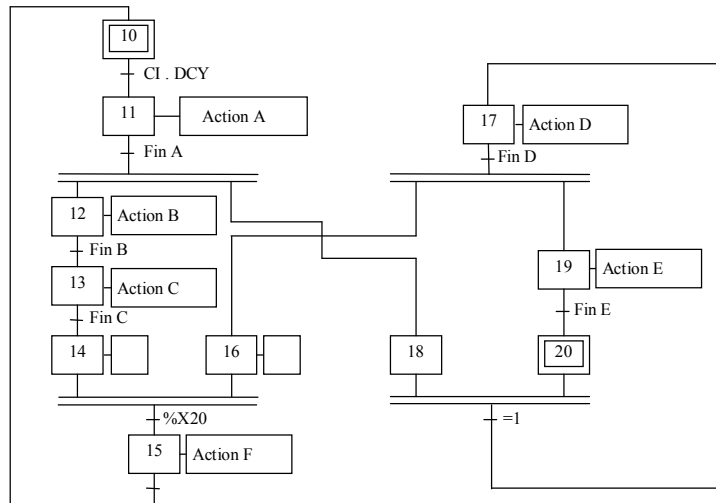


Fig. 10 Exemple de divergence et de convergence en plusieurs séquences simultanées

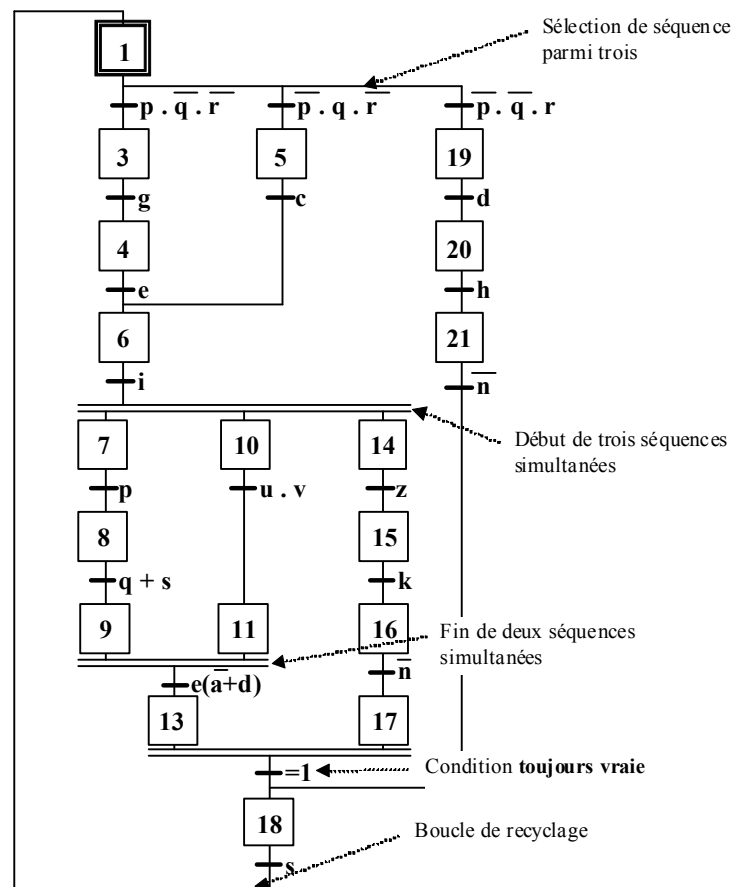


Fig. 10 Exemple de cycle décrit par un Grafcet

3. Règles de syntaxe :

L'alternance étape - transition et transition - étape doit toujours être respectée pour chaque séquence parcourue, par exemple :

- 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 toujours être séparées par une étape.

4. Ordres détaillés ou actions détaillées et conditions de transition détaillées

4.1 Qualification des ordres (actions)

La correspondance exacte entre la durée des ordres et la durée de l'activité des étapes est indiquée en affectant aux symboles des ordres détaillés les lettres suivantes :

- N (niveau de maintien)
- S (mémorisation d'activation) ;
- D (retardé) ;
- L (limité dans le temps).

Si la durée de L est courte, L peut être remplacée par :

- P (Impulsion sur un cycle de scrutation).

4.2 Ordres (actions) détaillé(e)s :

N°	Symbole	Description	Écriture Automgen
Action non mémorisée			
5.2.			
Action mémorisée			
5.3.			
Action non mémorisée mais retardée			
5.4.			
Action non mémorisée mais limitée dans le temps			
5.5.			

Action mémorisée et retardée			
5.6.			<p><i>Même si l'étape 18 n'est plus active l'action s'exécute après les délais</i></p>

Action retardée et mémorisée			
N°	Symbole	Description	Écriture Automgen
5.7.			
Action mémorisée et limitée dans le temps			
5.8.			

4.3 Ordres (actions) conditionnel(le)s :

Un ordre peut être soumis à une condition logique de validation avant ou après le traitement indiqué de son signal d'étape (section "a" du symbole d'ordre). Cela est particulièrement important

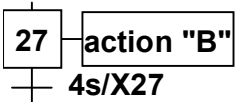
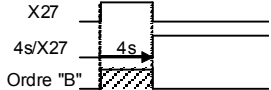
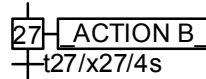
quand l'ordre est mémorisé. Une telle condition peut être indiquée à l'intérieur ou à l'extérieur du symbole d'ordre, suivant la place disponible pour l'écriture.

N°	Symbole	Description	Écriture Automgen
Action conditionnelle			
6.1.	<p>Forme 1</p>		
6.2.	<p>Forme 2</p>		
Action mémorisée et conditionnelle			
6.3.	<p>Forme 1</p>		
6.4.	<p>Forme 2</p>		
N°	Symbole	Description	Écriture Automgen
Action conditionnelle et mémorisée			
6.3.	<p>Forme 1</p>		
6.4.	<p>Forme 2</p>		

4.4 Conditions de transition détaillées :

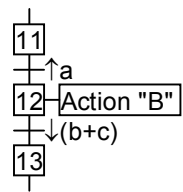
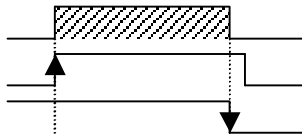
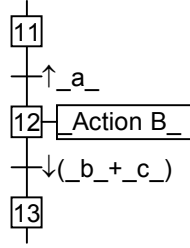
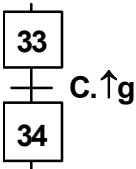
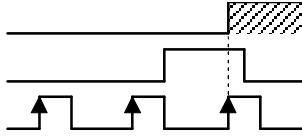
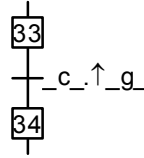
4.4.1 Annotation de la dépendance de temps :

N°	Symbole	Description	Écriture Automgen
----	---------	-------------	-------------------

7.1.		<p>Durée d'activité de l'étape 27 limitée à 4s par le franchissement de la transition</p> 	
------	---	---	---

4.4.2 Annotation de l'état logique

Le franchissement d'une transition peut dépendre non seulement de la présence de l'état logique de la variable logique symbolisant la condition mais aussi de son changement d'état logique

N°	Symbole	Description	Écriture Automgen
8.1	$\uparrow c$	Transition du signal logique binaire c de 0 à 1 (front montant)	
8.2.	C	État 1 du signal logique binaire c	
8.3.	$\downarrow c$	Transition du signal logique binaire c de 1 à 0 (front descendant)	
8.4.	\bar{c} ou non c	État 0 du signal logique c	
N°	Symbole	Description	Écriture Automgen
8.5.		<p>L'activation et la désactivation de l'étape 12 sont provoquées par des conditions de transition dynamiques.</p> 	
8.6.		<p>La transition de l'état actif de l'étape 33 vers l'étape 34 se produira uniquement lorsque la condition "c" est vraie pendant le temps et "g" passe de 0 à 1.</p> 	

Grafcet sous programme

La notion de Grafcet sous-programme ainsi que le concept de macro-étape associé font partie de ce qu'il est convenu d'appeler des macro – représentations.

Avec une macro – représentation on se donne le moyen de reporter à plus tard ou sur une autre page la description détaillée de certaines séquences du grafcet

La notion de sous-programme est empruntée au langage informatique. Des instructions spécifiques sont prévues pour appeler un sous-programme à partir d'un programme principal puis pour revenir à celui-ci lorsque le sous-programme est terminé. On rencontre ce type d'instructions dans les langages assembleurs des microprocesseurs et dans les langages évolués.

Le sous-programme informatique décrit une suite d'opérations qui seront exécutées plusieurs fois dans le même cycle. Il en sera de même du Grafcet sous-programme décrivant une suite d'actions. Celui-ci sera lancé et relancé plusieurs fois dans le même cycle de production.

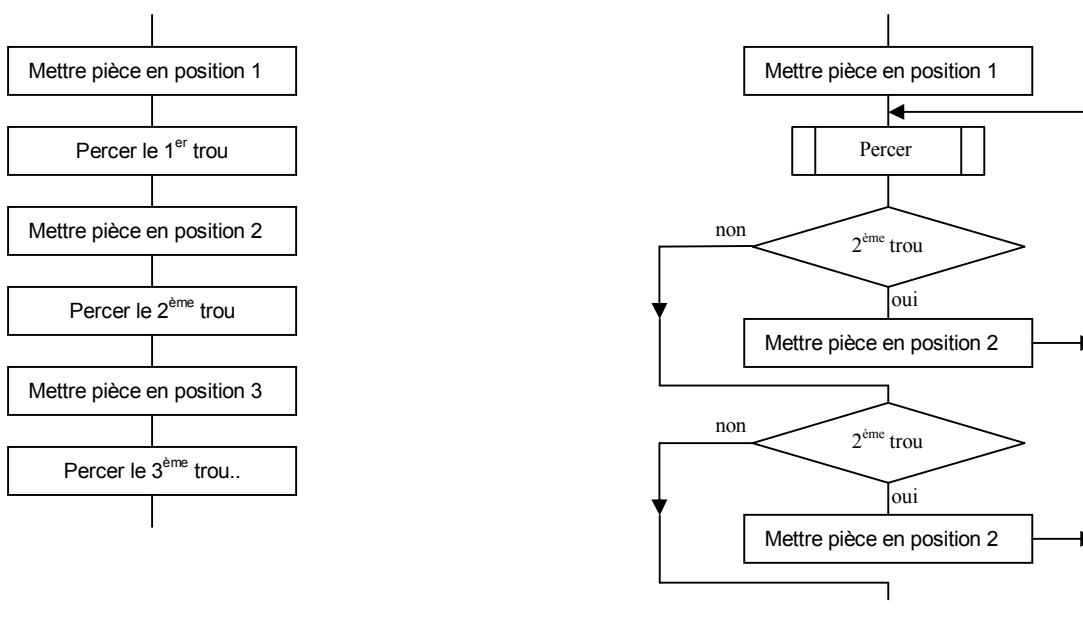
EXEMPLE

Un cycle d'usinage comportant le perçage de multiples trous. La séquence :

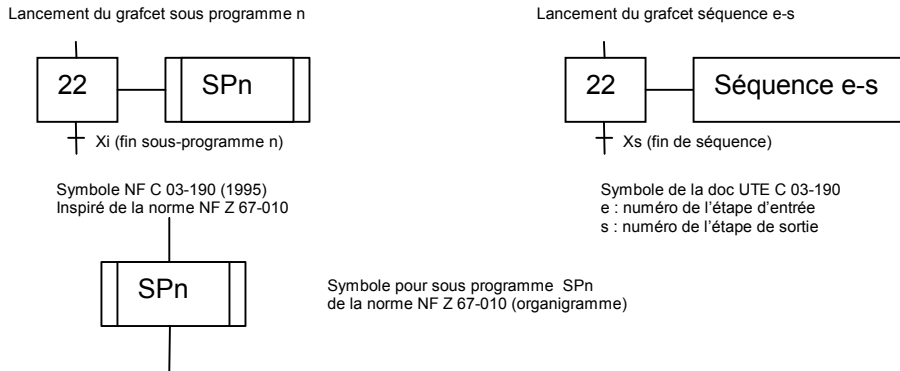
1. Mise en rotation de la broche
2. Descente d'approche en vitesse rapide
3. Descente perçage en vitesse lente
4. Remontée de la broche en vitesse rapide
5. Arrêt de la broche en rotation

Est à répéter autant de fois qu'il y a de trous à percer. Cette séquence peut être organisée en sous-programme.

Dans le cas d'une description par Grafcet, le Grafcet sous-programme sera appelé à chaque perçage par le Grafcet principal.



Symboles de la macro-représentation de lancement d'un Grafcet sous-programme



Structure d'un grafcet sous-programme (séquence répétitive)

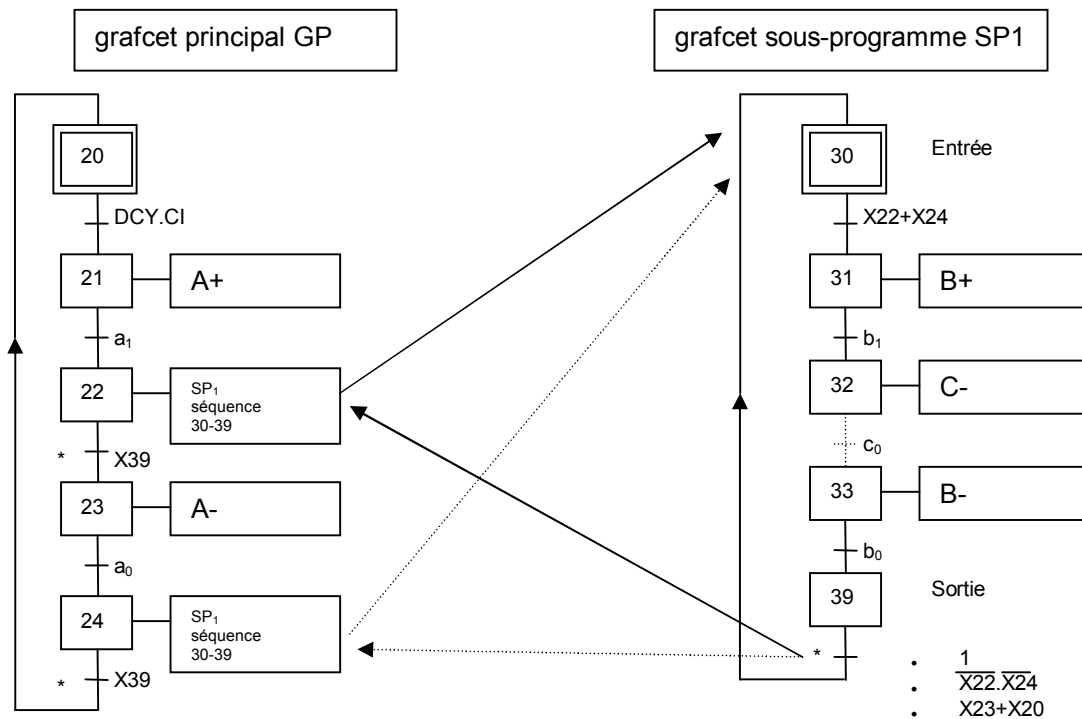
Cette structure est caractérisée :

Par son étape d'entrée jouant le rôle d'étape initiale indispensable pour le lancement du sous-programme,

Par son étape de sortie concrétisant la fin de l'exécution du sous-programme.

Aucune action extérieure ne doit être associée à ces deux étapes

Coordination entre le GRAFCET principal et le GRAFCET sous-programme



Interprétation

Le grafcet principal (GP) comporte les étapes de lancement 22 et 24 du grafcet sous-programme SP1. Ce grafcet (SP1) comporte une étape initiale d'attente ou d'entrée (étape 30) et une étape de sortie ou de retour au grafcet GP (étape 39).

L'activation de l'étape d'appel 22 autorise le franchissement de la transition $t_{30 \rightarrow 31}$. Il y a donc bien lancement du grafcet SP1. Ensuite le grafcet SP1 s'exécute jusqu'à l'activation de l'étape 39 de sortie ou de retour.

L'activation de l'étape 39 autorise le franchissement de la transition $t_{22 \rightarrow 23}$, le grafcet GP reprend le cours de ses évolutions. Simultanément, la transition $t_{39 \rightarrow 30}$ est franchie (condition de transition toujours vraie (=1)). Le grafcet SP1 se réinitialise. Les transitions franchies simultanément sont repérées par des astérisques (*).

La relance du Grafcet SP1 par l'étape 24 se fera suivant le même enchaînement : lancement du grafcet SP1, exécution, retour au Grafcet GP et réinitialisation du grafcet SP1.

Cette coordination entre GP et SP1 fonctionne suivant le principe dit **appel-réponse** :

Un ordre d'appel ou de lancement est émis par le grafcet principal en direction du grafcet sous-programme : conditions X22 et X24.

Une réponse est fournie au grafcet principal en fin d'exécution du grafcet sous-programme : condition X39.

Remarque : si le grafcet sous-programme n'est pas réinitialisé, son lancement est impossible. Il n'est donc pas nécessaire d'introduire la condition X30 dans les réceptivités du grafcet principal.

La réinitialisation du grafcet sous-programme peut être obtenue de deux manières :

Soit par une condition de transitions toujours vraie (=1) suivant solution UTE C 03-190.

Soit par contrôle de la désactivation des étapes de lancement (condition /X22./X24)

Soit par la contrôle de l'activation des étapes faisant suite aux étapes de lancement (condition X23 + X20)

La deuxième solution permet, avant de réinitialiser le g une macro-représentation de lancement d'un autre grafcet sous-programme (imbrication des grafcet sous-programme, de vérifier si la relance du grafcet principal est effective. Ceci est obtenu en contrôlant la désactivation de la ou des étapes de lancement du sous-programme. Sur le plan pratique, la condition de transition de sortie est obtenue en prenant le complément de la condition de transition d'entrée.

Un grafcet sous-programme peut lui-même contenir une macro-représentation de lancement d'un autre grafcet sous-programme (imbrication des sous programmes)

Remarque :

La notation proposée par la documentation UTE C 03-190 suppose le grafcet sous-programme déjà décrit par ailleurs puisqu'il faut indiquer le numéro de l'étape d'entrée (e) et le numéro de l'étape de sortie (s). Cette condition n'est pas remplie lorsqu'on pratique l'analyse descendante.

Grafcet de tâches

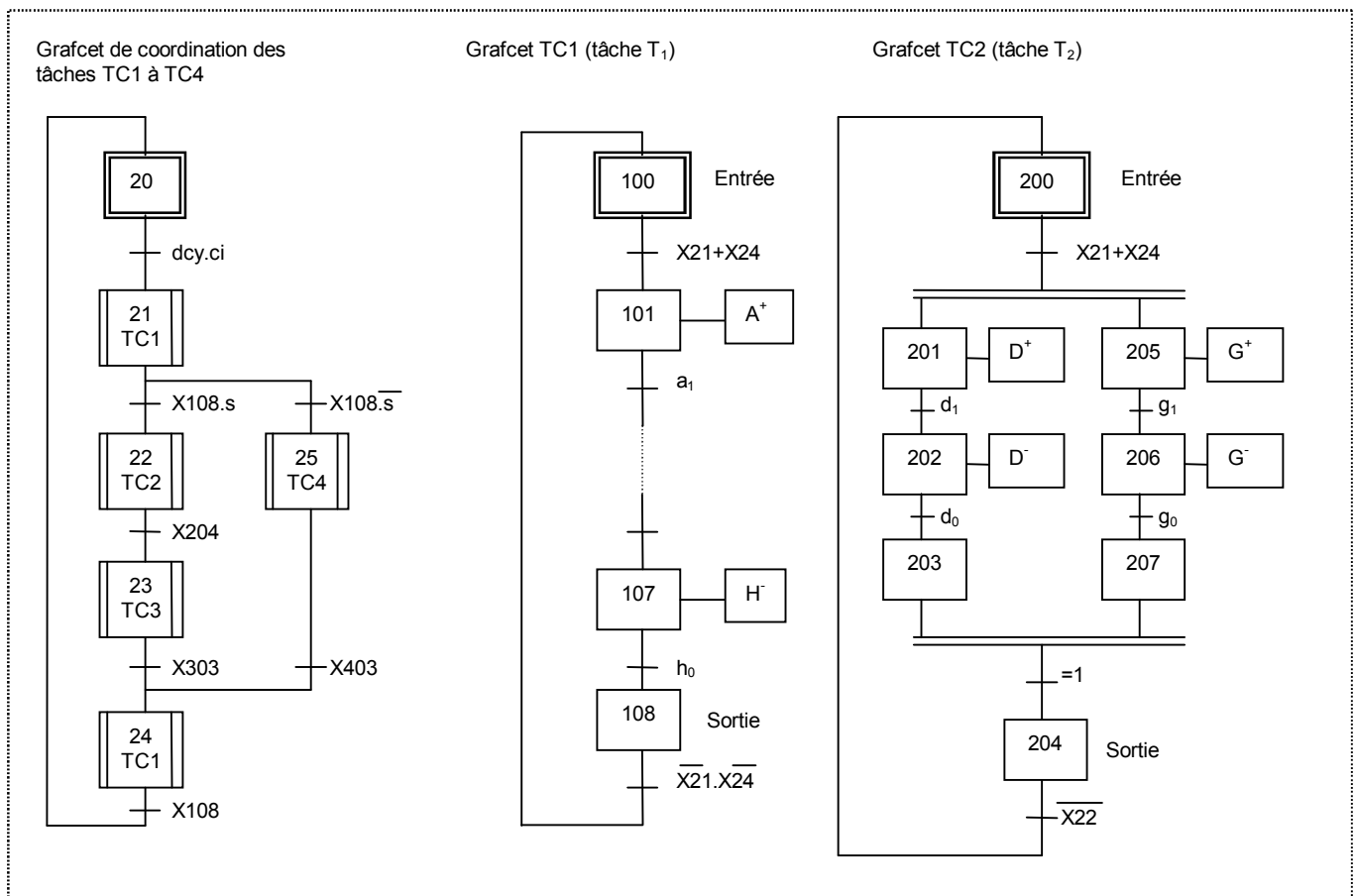
La tâche correspondant à la réalisation d'une fonction bien déterminée qui pourra n'être exécutée qu'une seule fois dans le cycle de production par opposition à un sous programme qui sera toujours exécuté plusieurs fois.

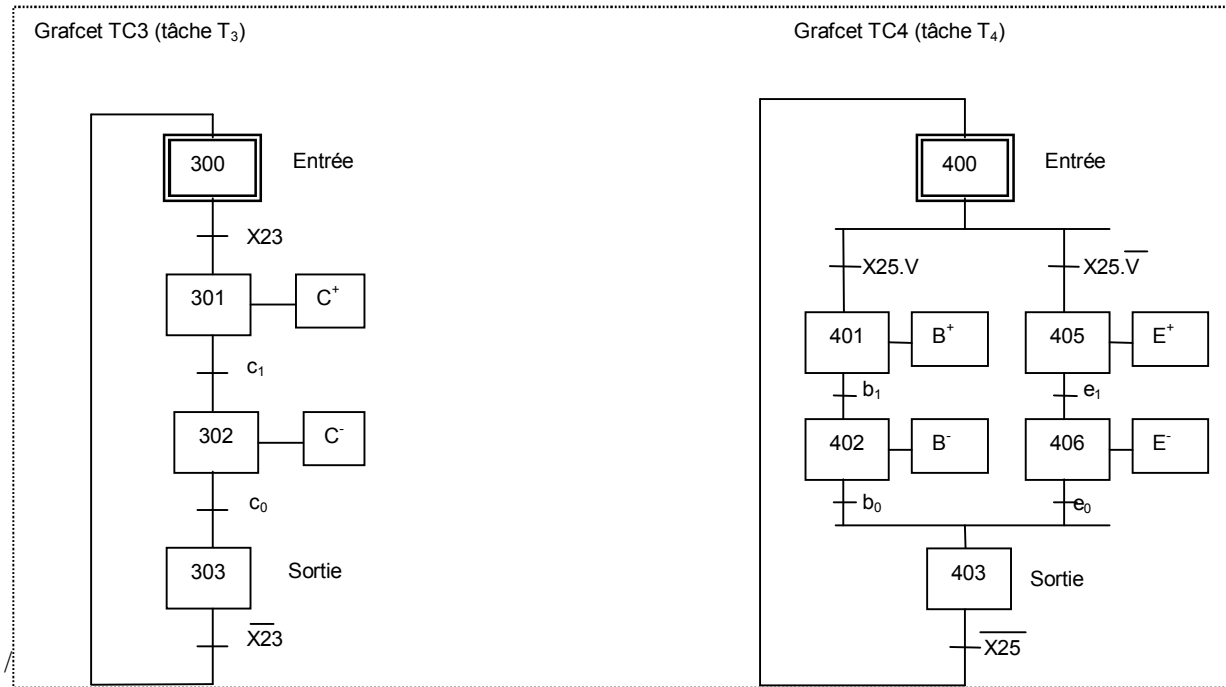
Exemple de tâches :

- Dosage d'un produit
- Lancement des moteurs
- Marche de préparation (état F2 du Gemma)
- Marche de clôture (état F3 du Gemma)

Il n'y a aucune différence de structure et de principe de lancement entre un grafcet de tâche et un grafcet sous-programme.

Un grafcet de coordination des tâches est décrit au niveau 1 de l'analyse descendante. Dans sa forme définitive au niveau 2, ce grafcet lancera les différentes tâches, décrites chacune indépendamment, par un grafcet de tâche comme le montre la figure suivante.





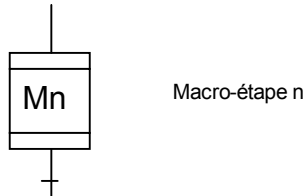
Macro- étape et expansion

Définition de la macro-étape

La macro-étape est une représentation unique d'un ensemble d'étapes et de transitions appelée « expansion de la macro-étape ».

La macro-étape se substitue à une étape normale du grafcet. L'expansion peut toujours s'insérer en lieu et place de la macro-étape.

La macro-étape est symbolisée par un rectangle divisé en trois parties par deux traits horizontaux :



La macro-étape est repérée à l'intérieur (ou à l'extérieur) de la case centrale du rectangle par un identificateur numérique ou alphanumérique. Les autres cases pourront être utilisées pour un repérage complémentaire, par exemple le numéro des étapes d'entrée et de sortie lorsque l'expansion associée aura été définie.

NB : la macro-étape n'a pas d'existence physique et ne peut être assimilée à une étape sinon les règles d'évolution du grafcet ne seraient pas respectées.

En effet comme on associe, le plus souvent, à la transition aval une condition de transition toujours vraie, la transition devrait être franchie dès que la macro-étape est activée. Or ce n'est justement pas le cas puisque l'expansion n'a pas été exécutée (voir la suite). Il s'ensuit également qu'aucune action ne peut être associée à une macro-étape.

Structure de l'expansion de la macro-étape

Cette structure obéit à certaines règles de construction, à savoir :

L'expansion a toujours une étape d'entrée E (ou IN) et une étape de sortie S (ou out) ;

Le franchissement de la transition amont de la macro-étape déclenche l'activation de l'étape d'entrée de l'expansion

L'activation de l'étape de sortie de l'expansion déclenche le franchissement de la transition aval de la macro-étape, c'est pourquoi la condition de transition généralement associée à cette transition est toujours vraie (=1)

L'étape d'entrée est une étape normale qui une fois activée assure le début d'évolution de l'expansion ; elle ne peut jamais être une étape initiale ; on peut lui associer une action.

L'étape de sortie sert à concrétiser le fin de l'exécution de l'expansion, on ne doit pas lui associer d'action externe à cause de l'instabilité liée à la réceptivité =1 du grafcet de gestion.

l'expansion peut comporter des étapes initiales pour pouvoir, notamment, franchir l'entrée ou la sortie d'un parallélisme structural interne.

une expansion de macro-étape peut elle-même contenir des macro-étapes.

une expansion peut comporter tous les types de structures propres au grafcet : convergence en et ou en ou, divergence en et ou en ou.

l'expansion possède trois états :

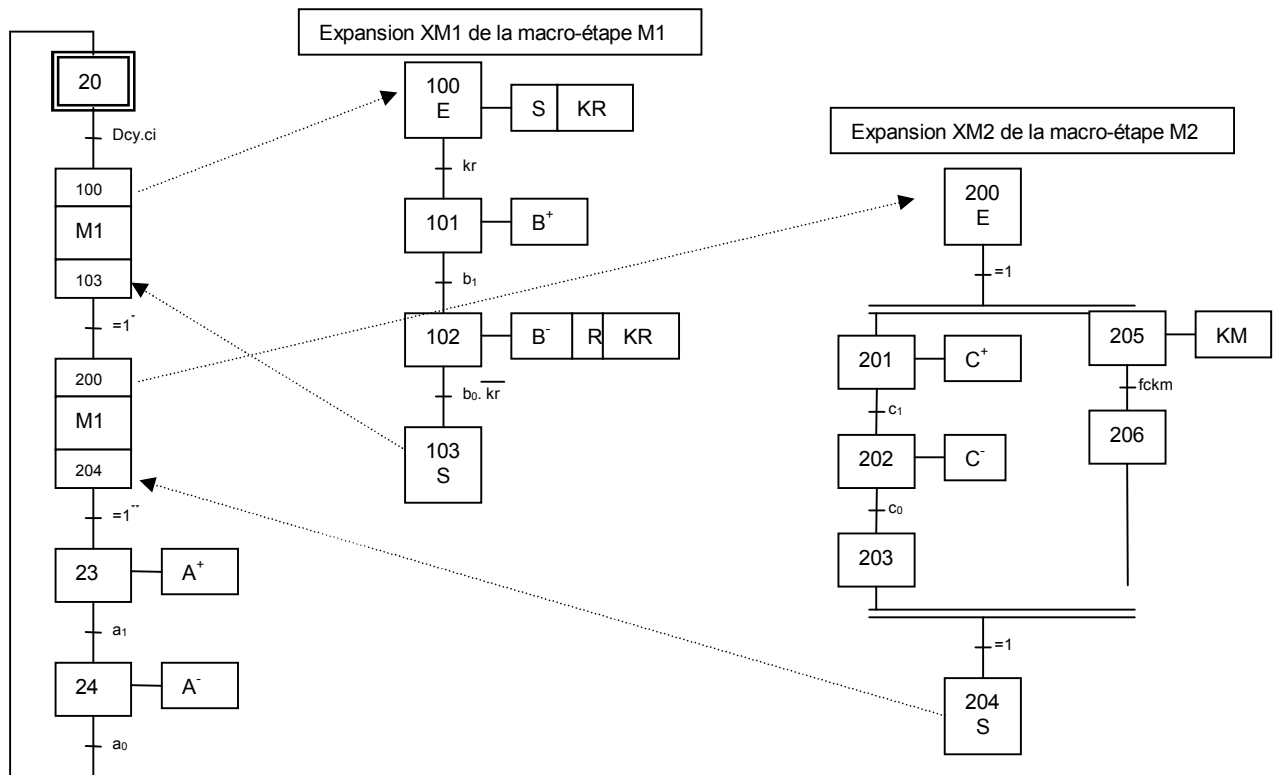
état repos : aucune étape active

état de fonctionnement : évolution en cours

état final : étape de sortie active.

Conditions d'évolutions de l'expansion de la macro étape

L'exemple de la figure suivante montre des expansions dans étapes initiales. Si on se trouve dans les situations où les étapes 20 ou 23 ou 24 sont actives alors on dit que les expansions se trouvent dans une situation vide.



Au niveau des réceptivités 1* et 1** on peut choisir entre différentes possibilités telles que :

1* = 1 ou X103 ou XSM1 ou XOUTM1

1** = 1 ou X204 ou XSM2 ou XOUTM2

Commentaire :

L'expansion XM1 en attente est au repos (situation vide). Le franchissement de la transition amont de la macro-étape M1 active l'étape 100 de l'expansion XM1 et désactive l'étape 20 du grafcet principale. La macro-étape est active. L'expansion XM1 évolue alors de façon autonome.

En fin d'évolution de l'expansion XM1, l'activation de l'étape 103 valide la transition aval de la macro-étape M1. Le franchissement de cette transition active la macro-étape M2 et désactive l'étape 103 de l'expansion XM1. L'expansion XM1 se trouve de nouveau en situation vide. La macro-étape M1 est désactivée.

Le même processus est à mettre au compte de la macro-étape M2. Son expression XM2 est lancée et évolue jusqu'à l'étape 204. La transition M2 vers 23 est franchie et l'expansion XM2 se retrouve en situation vide.

Remarque : une séquence ou une tâche devant être exécutée n fois dans le même cycle nécessite n macro-étapes distinctes associées à autant d'expressions de même structure mais composées d'étapes différentes

Intérêt et limites des macro-représentations

L'analyse descendante d'un système permet d'affiner progressivement la connaissance du système automatisé de production, objet de l'étude.

Le découpage en tâches ou en sous-programme permet de mieux structurer l'ensemble de la description par grafcet. La macro-représentation permettent de ne pas d'encombrer de détails inutiles dès le début de l'analyse et de se consacrer uniquement à l'analyse des tâches et de leur coordination.

La subdivision d'un programme en des tâches et des sous programme permet, d'une part de simplifier l'analyse et la mise en œuvre d'un programme et d'autre elle permet de simplifier l'intervention au niveau des sous programme afin d'améliorer le fonctionnement du système. En plus, dans le cas d'une réalisation programmée, l'emploi de sous-programmes pour les tâches ou les séquences répétitives permet un gain appréciable de place en mémoire.

En conclusion, le concept de grafcet sous-programme ou de tâche présente l'avantage d'être indépendant du langage de l'API. Il est donc implantable sur tous les types d'automates programmables industriels. De plus, par rapport au concept de macro-étape, il apporte un gain en espace mémoire important dans le cas de séquences ou de tâches répétitives.

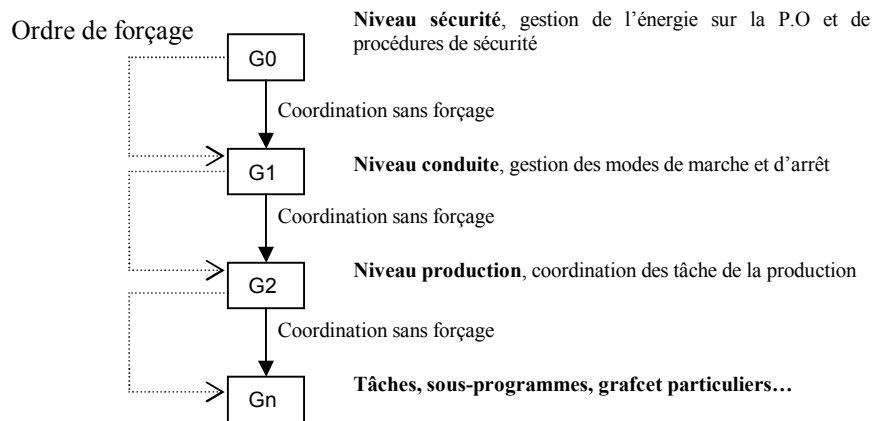
Forçage de situation

Définition du forçage de situation

L'ordre de forçage de situation émis par un grafcet partiel permet d'imposer une situation à un autre grafcet partiel.

La situation du grafcet forcé évolue alors sans franchissement de transition de sa situation courante à la situation forcée.

L'ordre de forçage ne peut être émis que par une étape d'un grafcet partiel hiérarchiquement supérieur pour modifier la situation d'un grafcet partiel hiérarchiquement inférieur. Il est donc indispensable de structurer la description de manière à établir une hiérarchie entre les grafctet.



Situation courante forcée dans une situation précisée non vide

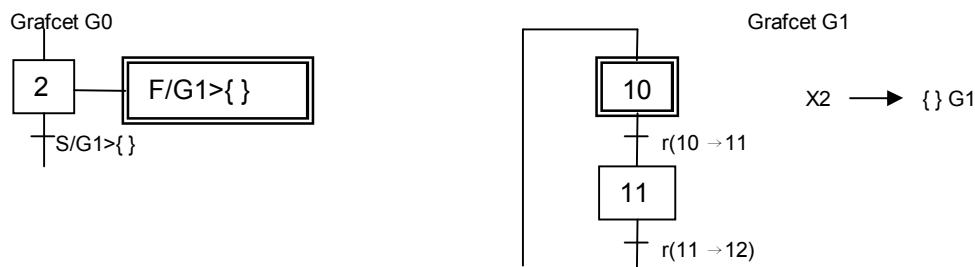


L'activation de l'étape 1 du grafctet G0 force le grafctet G1 sur la situation non vide $s=\{10\}$ de ce grafctet. L'étape 10 est activée (forcée à 1), les autres étapes du grafctet G1 sont désactivées (forcées à 0).

Notation du forçage d'une situation non vide :

- De l'action de forçage : F/grafctet à forcer > {situation forcée}
- Du grafctet forcé : étape de forçage → accolé à la (ou les) étapes à forcer à 1

Situation courante forcée dans une situation vide



Commentaire

L'activation de l'étape 2 du grafcet G0 force le grafcet G1 sur la situation vide $s=\{ \}$ de ce grafcet. Toutes les étapes du grafcet G1 sont désactivées y compris les étapes initiales.

Notation du forçage d'une situation vide :

- De l'action de forçage : $F/\text{grafcet à forcer} > \{ \}$
- Du grafcet forcé : $\text{étape de forçage} \rightarrow \{ \}$ grafcet à forcer

Ordre de forçage.

Règles d'évolution par forçage

- Un grafcet n'est peut être forcé que par un grafcet hiérarchiquement supérieur.
- Un grafcet inférieur ne peut être forcé que dans une seule situation à la fois, à partir d'un ou plusieurs grafcet supérieurs.
- L'ordre de forçage est prioritaire sur les autres conditions assurant l'évolution du grafcet forcé.
- Lorsqu'il s'agit d'un forçage de situation non vide, l'ordre de forçage provoque simultanément l'activation des étapes correspondant à la situation imposée et la désactivation des autres étapes du grafcet forcé.

Exemple :

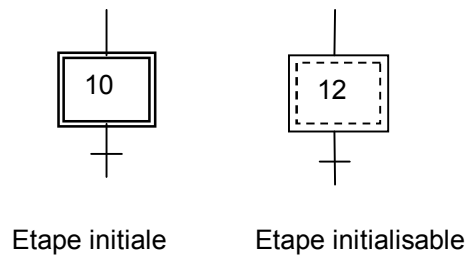
$F/G3 > \{30\}$ est un ordre de forçage du grafcet G3 dans la situation imposée $s=\{3\}$. Dès que l'étape, à laquelle cet ordre est associé, devient active, cet ordre est obligatoirement émis. Il provoque simultanément l'activation de l'étape 30 et la désactivation des autres étapes du grafcet G3.

- Lorsqu'il s'agit d'un forçage de situation vide, l'émission de l'ordre de forçage provoque simultanément la désactivation de toutes les étapes du grafcet désigné.

Remarque :

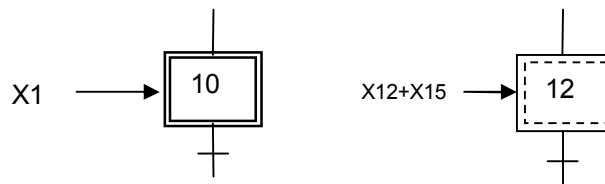
Dans certains cas on peut simplifier l'écriture des forçages :

- *Si la situation forcée ne prête à aucune confusion. Exemple : $F/ > \{10, 20\}$ force la situation $s=\{10, 20\}$*
- *Si le forçage s'applique à plusieurs grafcet. Exemple : $F/G1, G2 > \{10, 20\}$ force le grafcet G1 dans la situation $\{10\}$ et le grafcet G2 dans la situation $\{20\}$.*
- *Si le forçage entraîne la réinitialisation d'un ou plusieurs grafcet. Exemple $F/G1 > \{INIT\}$ force G1 dans sa situation initiale. $F/G1-G5 > \{INIT\}$ force G1, G2, G3, G4, G5 dans leurs situations initiales.*



Les étapes correspondant à la situation forcée et non vide, si elles ne sont pas déjà des étapes initiales, sont par définition des étapes initialisables.

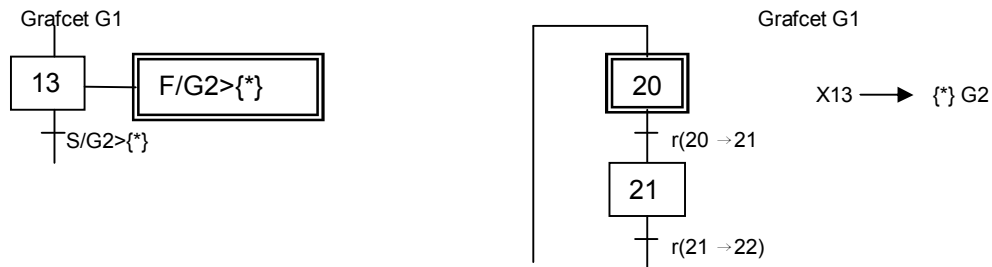
On les distinguera des étapes initiales en représentant le cadre intérieur avec des pointillés ou des tirets courts. On peut rappeler, par une flèche annotée placée à gauche de l'étape initiale ou initialisable, l'origine du forçage à 1 de cette étape, étant sous-entendu que les autres étapes sont forcées à 0.



Figeage de situation

Introduction : le figeage d'une situation ne présente par le forçage au vrai sens du mot du grafcet mais au contraire il consiste à bloquer l'évolution du grafcet dans une situation choisie à l'avance : situation courante ou situation prédéterminée.

Situation figée dans la situation courante

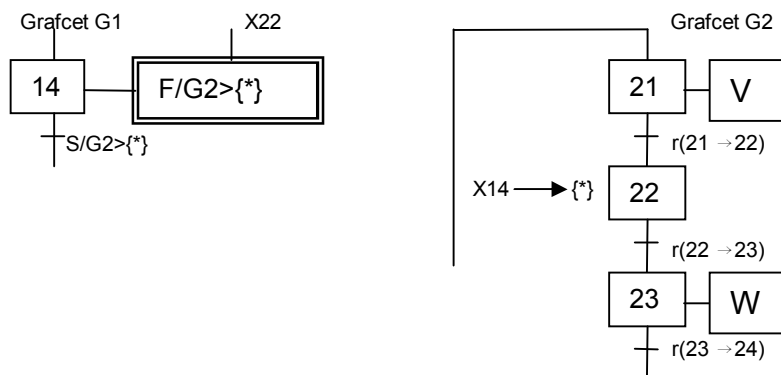


Commentaire : l'activation de l'étape 13 du grafcet G1 fige la situation du grafcet G2 dans sa situation courante. Les étapes actives restent activées, les étapes inactives restent désactivées.

La notation du figeage dans la situation courante :

- De l'action de figeage : F/grafcet à figer > {*}
- Du grafcet à figer : ordre de figeage → {*} grafcet à figer

Situation évoluant vers une situation de figeage prédéterminée.



La situation du grafcet G2 continue d'évoluer jusqu'à la situation s={22} où elle se figera. Il s'agit en fait d'un figeage conditionné par la situation de figeage à atteindre. Aucune action ne doit être associée à cette étape.

La notation du figeage dans la situation prédéterminée :

- De l'action de figeage : F/grafcet à figer > {*} si situation de figeage est atteinte
- Du grafcet à figer : ordre de figeage → {*} accolé à l'étape où se fera la figeage.

Problème entraînés par le figeage de situation

Cas du figeage dans la situation courante

Le figeage du grafcet de production normale dans sa situation actuelle (situation courante) sur ordre de l'opérateur ou à la suite d'un incident, présente l'avantage de permettre la reprise de la production sans procédure compliquée. Il suffit de quitter l'étape émettant le signal de figeage. La désactivation de cette étape annule l'ordre de figeage du grafcet de production normale et la production reprend là où elle s'était arrêtée.

Le figeage obtenu, les ordres associés aux étapes actives des grafcet de tâches continuent d'être émis. Si ces ordres sont destinés à des entrées de préactionneurs électriques, que va-t-il se passer ?

Les bobines des contacteurs et des distributeurs vont rester sous tension pendant toute la période de figeage. Les moteurs, par exemple, vont continuer à tourner, ce qui n'est généralement pas souhaitable sur le plan de la sécurité.

Comment résoudre ce problème ?

Il faut émettre, en même temps que l'ordre de figeage du grafcet de production, un ordre de forçage à l'état 0 des sorties de la partie opérative.

Le forçage à 0 des sorties de la PO, en fonction de la sécurité souhaitée, peut entraîner :

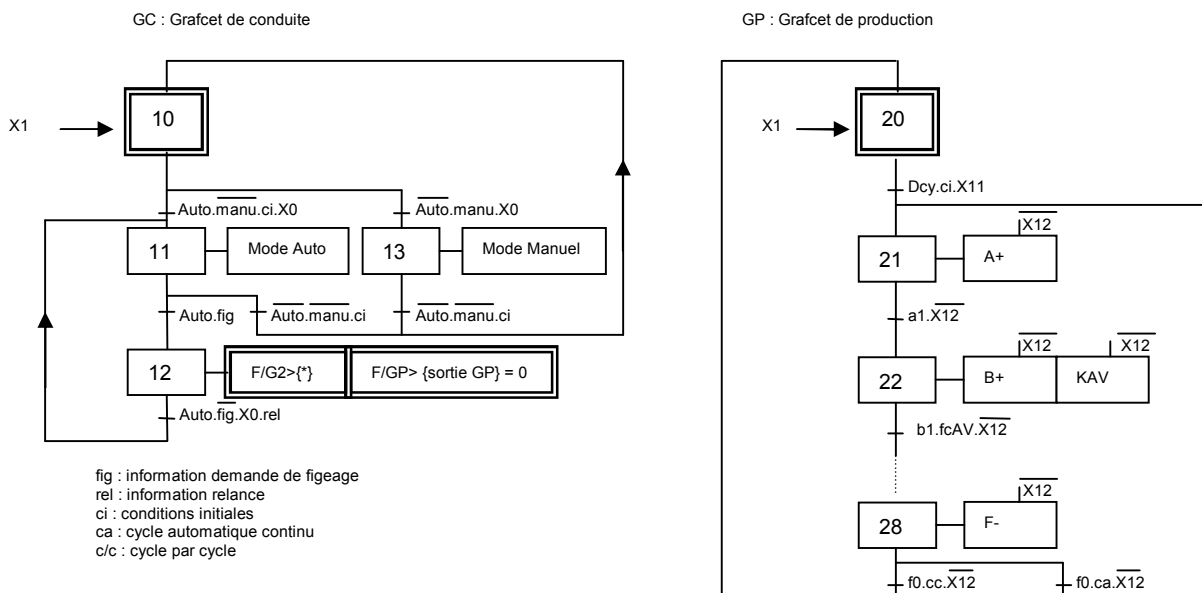
- Soit l'arrêt sur place des actionneurs,
- Soit l'arrêt après terminaison des mouvements en cours (cours des mécanismes actionnés par des vérins ou des moteurs, etc.)

Une des solutions consiste d'une part à annuler les ordres émis par les grafcet de tâches et d'autre part à choisir des actionneurs et des préactionneurs de technologie appropriée : moteur-frein, vérin bloqueur de tige...

L'annulation des ordres de sortie peut être obtenue :

- Soit par forçage à zéro des entrées des préactionneurs (dans le grafcet de l'exemple suivant le forçage à 0 est fait par la condition $\overline{X12}$ associée aux actions).
- Soit par le câblage d'un contact \overline{fig} (normalement fermé) lié au capteur-opérateur FIG de demande de figeage dans le circuit d'alimentation des entrées des préactionneurs ou des cartes de sorties de l'API (sécurité de premier niveau)
- Soit en exploitant un bit système approprié (%S23 pour les automates TSX de Schneider)
- Soit en combinant les procédés afin d'obtenir une redondance améliorant la sécurité.

Exemple (avec grafcet de conduite simplifié)

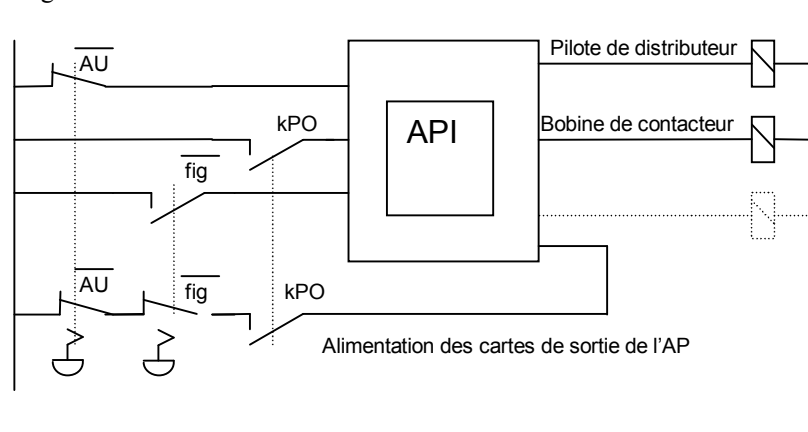


Commentaire

Le figeage du grafcet de production de l'exemple est obtenu par la mise à zéro des réceptivités si l'étape 12 est active. Les vérins A,B, ..., F terminent leurs courses, le moteur s'arrête immédiatement. Il est sous-entendu qu'un choix approprié des actionneurs et des préactionneurs a été fait.

Exemple : solution câblée du forçage à 0 des sorties

AU : arrêt d'urgence
 Fig : demande de figeage
 kPO : relais électromagnétique de mise en énergie de la P.O



Commentaire

Dans cette solution, l'action sur le coup de poing FIG ou sur le coup de poing AU entraîne, dans les deux cas, la coupure de l'alimentation des cartes de sorties. Par suite, les entrées des préactionneurs ne sont plus alimentées.

Cas de figeage de situation avec mise en situation vide

Il est possible de figer le grafcet de production normale et les grafcet de tâches qu'il supervise en procédant de cette manière :

Sur ordre de figeage, mémorisation de la situation courante, puis mise des grafcet de production dans une situation vide F/Gp>{ }. Les ordres associés aux étapes de ces grafcet sont ainsi systématiquement annulés.

Pour relancer l'automatisme, il faut forcer les grafcet concernés sur la situation sauvegardée. Cette procédure suppose qu'on dispose des moyens appropriés pour la réaliser. Le logiciel de programmation assistée automgen de la société IRAI ainsi que le langage PL7-3 V5 de Schneider possèdent des instructions autorisant ce type de procédure.