

Cours Complet sur le Grafset & Exercices Corrigés



Chapitre1 : LE GRAFCET

Table des matières

LE GRAFCET	4
1 Introduction.....	6
1.1 Définition	6
1.2 Domaine d'application.....	6
2 Langage et interprétation	6
2.1 Principe du grafcet	6
2.2 Règles d'évolution du grafcet.....	8
2.3 Actions associées aux étapes.....	10
2.3.1 Ordre continu.....	10
2.3.2 Ordre conditionnel.....	10
2.3.3 Ordre de mémorisation de l'action.....	10
2.3.4 Ordre retardé (D)	11
2.3.5 Ordre de durée limitée (L).....	12
2.3.6 Décompteur	12
2.4 Transitions.....	13
2.4.1 Transition.....	13
2.4.2 Receptivité.....	13
2.4.3 Liaison orientée	13
2.5 Séquences de base	13
2.5.1 Grafcet linéaire (séquence unique).....	13
2.5.2 Divergence et convergence en "OU".....	14
2.5.3 Divergence et convergence en "ET"	15
2.5.4 Sauts d'étapes et reprise de séquences	16
2.5.5 Sous-programmes	17
a) Macro-étape	17
b) Graphe auxiliaire	18
2.5.6 Traitement d'arrêt.....	20
a) sans séquence d'urgence "Figé"	20
b) séquence d'urgence	21
2.5.7 Traitement de modes de marche.....	22
a) Marche cycle par cycle	22
b) Cycle unique.....	23
c) Marche automatique / arrêt cycle	24
d) Marche auto/cycle par cycle.....	25

Table des illustrations :

Figure 1 : Principe du grafcet.....	7
Figure 2 : Etape initiale.....	8
Figure 3 : Franchissement d'une transition	8
Figure 4 : Evolution des étapes actives.....	9
Figure 5 : ordre continu.....	10
Figure 6 : ordre conditionnel.....	10
Figure 7 : ordre de mémorisation.....	11
Figure 8 : Ordre retardé.....	11
Figure 9 : ordre de durée limitée.....	12
Figure 10 : Décompteur	12
Figure 11 : Grafcet linéaire	13
Figure 12 : Divergence et convergence en "OU"	14
Figure 13 : réceptivités exclusives.....	15
Figure 14 : divergence et convergence en ET.....	15
Figure 15 : saut d'étapes.....	12
Figure 16 : reprise de séquence.....	16
Figure 17 : macro étape.....	17
Figure 18 : graphe auxiliaire	18
Figure 19 : graphe auxiliaire	19
Figure 20 : traitement d'arrêt figé.	20
Figure 21 : traitement d'arrêt d'urgence séquentiel.	21
Figure 22 : Marche cycle par cycle.....	22
Figure 23 : Marche cycle unique	23
Figure 24 : Marche automatique / arrêt cycle	24
Figure 25 : Marche auto / cycle par cycle.....	25

LE GRAFCET

1 Introduction

1.1 Définition

Le diagramme fonctionnel ou **GRAFCET** (Graphe de Commande Etape – Transition) est un moyen de description du cahier des charges d'un automatisme.

C'est une méthode de représentation graphique qui décrit les comportements successifs de la partie commande d'un système automatisé (ordres à émettre, actions à effectuer, événements à surveiller).

1.2 Domaine d'application

Le diagramme fonctionnel est indépendant des techniques séquentielles "tout ou rien", pneumatique, électrique ou électronique, câblées ou programmées, pouvant être utilisées pour réaliser l'automatisme de commande. Mais l'utilisation de séquenceurs, d'une part, et d'automates à instructions d'étapes d'autre part, permet une transcription directe du diagramme fonctionnel.

Cette représentation graphique concise et facile à lire est aisément compréhensible par toute personne en relation avec le système automatisé, du concepteur à l'utilisateur sans oublier l'agent de maintenance.

Utilisé industriellement, le GRAFCET est aussi enseigné dans les options techniques et l'enseignement supérieur.

Depuis les premières publications le concernant et surtout depuis la norme française NF C03-190 de 1982, cet outil a été travaillé et enrichi par le groupe systèmes logiques de l'**AFCET** (Association Française pour la Cybernétique Economique et Technique)

Il existe une documentation et symboles graphiques, diagramme fonctionnel "Grafcet" éditée par l'Union Technique de l'Electricité. UTE C03-190 Nov. 1990

2 Langage et interprétation

2.1 Principe du grafcet

Pour visualiser le fonctionnement de l'automatisme, le **GRAFCET** utilise une succession alternée d'**ETAPES** et de **TRANSITIONS**.

A chaque étape correspond une ou plusieurs actions à exécuter. Une étape est soit active, soit inactive. Les actions associées à cette étape sont effectuées lorsque celle-ci est active.

Les transitions indiquent avec les **LIAISONS ORIENTEES**, les possibilités d'évolution entre étapes.

A chaque transition est obligatoirement associée une condition logique pouvant être vraie ou fausse.

Cette condition de transition est appelée **RECEPTIVITE**. L'évolution d'une étape à une autre ne peut s'effectuer que par le franchissement d'une transition.

Une transition ne peut être franchie, donc activer l'étape suivante que :

- si elle est validée par l'étape antérieure active,
- et que les conditions de réceptivité soient satisfaites.

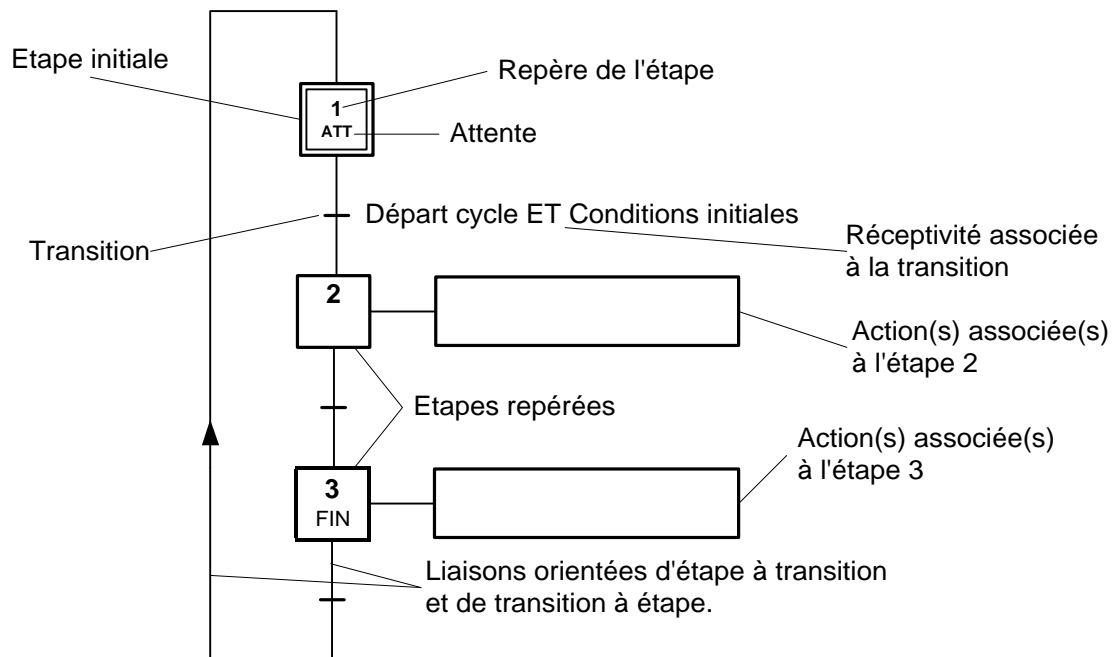


Figure 1 : Principe du grafcet

- Etape initiale** : représente une étape qui est active au début du fonctionnement. Elle se différencie de l'étape en doublant les côtés du carré.
- Transition** : la transition est représentée par un trait horizontal
- Réceptivité** : les conditions de réceptivité sont inscrites à droite de la transition
- Etape** : chaque étape est représentée par un carré repéré numériquement
- Action(s)** : elles sont décrites littéralement ou symboliquement à l'intérieur d'un ou plusieurs rectangles reliés par un trait à la partie droite de l'étape.
- Liaisons orientées** : indique le sens du parcours.

Niveaux d'emploi du GRAFCET

Afin de définir correctement le cahier des charges d'un équipement, le diagramme fonctionnel est utilisé à 2 niveaux :

- Niveau 1** : ne prend en compte que l'aspect fonctionnel du cahier des charges. Il ne considère que les actions à réaliser et les informations nécessaires pour les obtenir, sans spécifier comment elles seront technologiquement obtenues.
- Niveau 2** : pourra être différent du grafcet de niveau 1 compte tenu de la nature et en particulier de la technologie des capteurs et actionneurs utilisés.

2.2 Règles d'évolution du Grafcet

Règle 1 L'initialisation précise les étapes actives au début du fonctionnement. Elles sont activées inconditionnellement et repérées sur le GRAFCET en doublant les côtés des symboles correspondants.

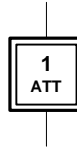


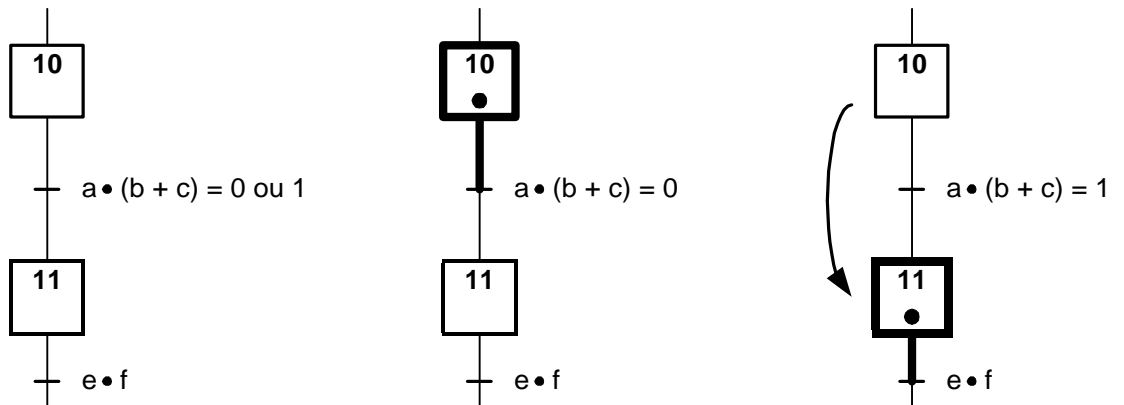
Figure 2 : Etape initiale

Règle 2 Une transition est soit validée soit non validée. Elle est validée lorsque toutes les étapes immédiatement précédentes sont activées.

Elle ne peut être franchie que:

- lorsqu'elle est validée
- et que la réceptivité associée à la transition est vraie.

La transition est alors obligatoirement franchie.



Transition non validée

La transition 10-11 est non validée, l'étape 10 étant inactive

Transition validée

La transition 10-11 est validée, l'étape 10 étant active, mais ne peut être franchie car la réceptivité $a \bullet (b + c) = 0$

Transition franchie

La transition 10-11 est franchie car la réceptivité $a \bullet (b + c) = 1$ L'étape 11 est active

Figure 3 : Franchissement d'une transition

Règle 3 Le franchissement d'une transition entraîne l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes. Cette évolution du GRAFCET est donc synchrone. Il y a évolution asynchrone lorsque le franchissement de la transition entraîne l'activation des étapes suivantes et que c'est la vérification de cette activation qui autorise la désactivation des étapes précédentes.

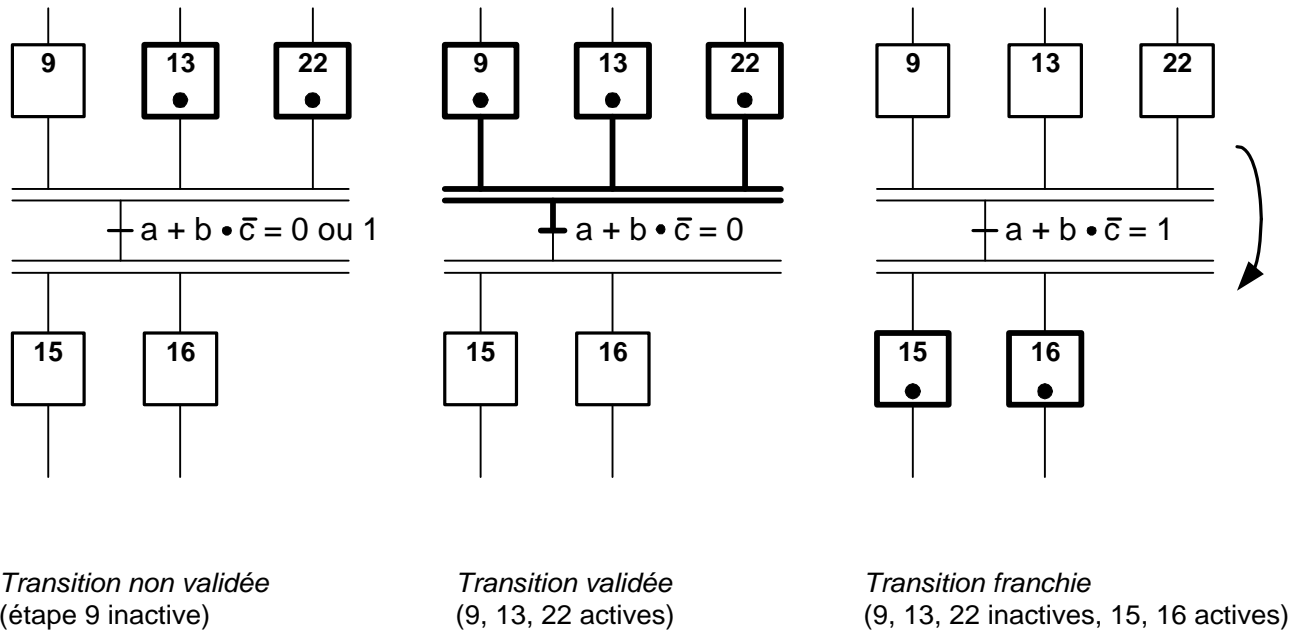


Figure 4 : Evolution des étapes actives

Règle 4 Plusieurs transitions simultanément franchissables sont simultanément franchies.

Règle 5 Si au cours du fonctionnement, une même étape doit être désactivée et activée simultanément, elle reste activée. L'activation doit être prioritaire sur la désactivation au niveau d'une même étape.

Remarque : La durée de franchissement d'une transition ne peut jamais être rigoureusement nulle, même si, théoriquement (règles 3 et 4), elle peut être rendue aussi petite que possible. Il en est de même de la durée d'activation d'une étape. En outre, la règle 5 se rencontre très rarement dans la pratique. Ces règles ont été ainsi formulées pour des raisons de cohérence théorique interne au GRAFCET.

2.3 Actions associées aux étapes.

Rappelons que l'ordre conditionne l'action.

2.3.1 Ordre continu

L'ordre est émis de façon continue tant que l'étape à laquelle il est associé est active

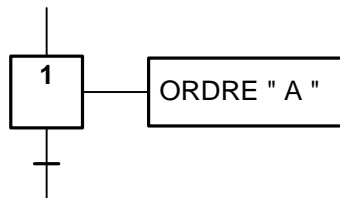


Figure 5 : ordre continu

2.3.2 Ordre conditionnel

L'ordre est émis lorsqu'en plus de l'activité de l'étape à laquelle il est associé, une condition logique spécifiée doit être satisfaite.

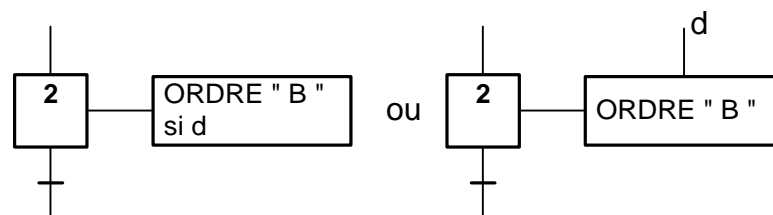


Figure 6 : ordre conditionnel

2.3.3 Ordre de mémorisation de l'action

Ces deux ordres, de mémorisation et d'effacement permettent d'élaborer l'action de sortie du composant :

- ordre de début d'action , noté : "action = 1" (set)
- ordre de fin d'action, noté : "action=0" (reset)

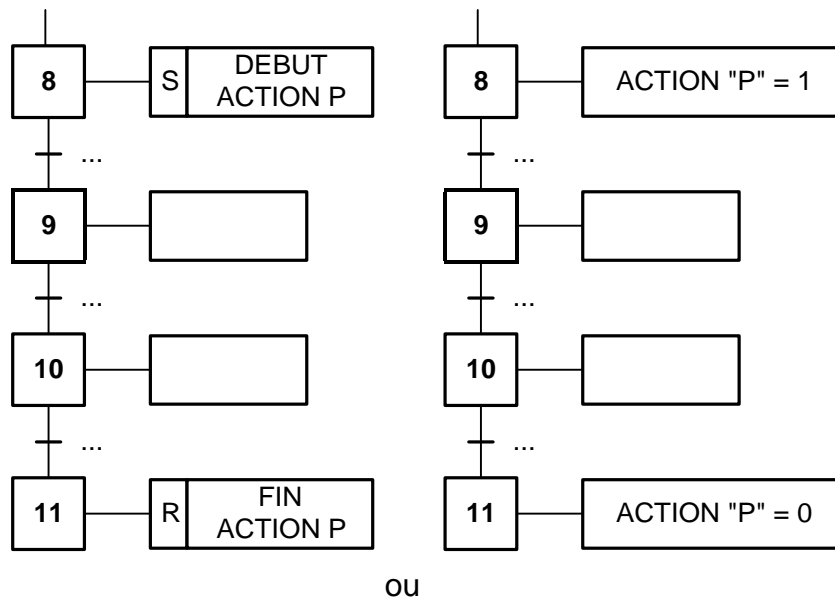


Figure 7 : ordre de mémorisation

2.3.4 Ordre retardé (D)

C'est un cas particulier d'un ordre conditionnel ou le temps intervient comme condition logique

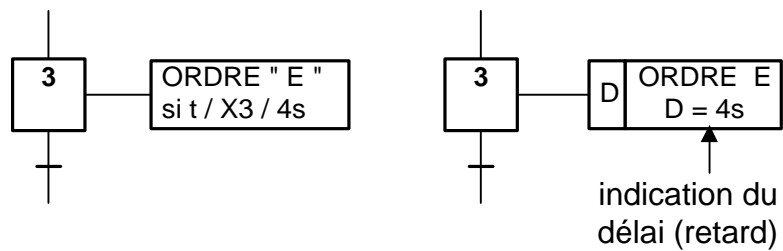


Figure 8 : Ordre retardé

2.3.5 Ordre de durée limitée (L)

L'ordre est émis immédiatement dès l'activation de l'étape à laquelle il est associé, mais sa durée est limitée à la valeur spécifiée

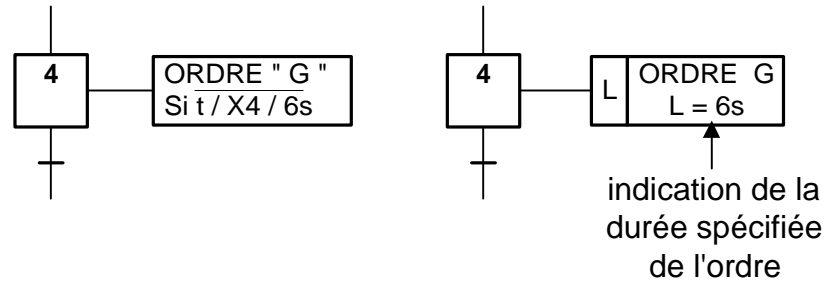


Figure 9 : ordre de durée limitée

2.3.6 Décompteur

Il faut prévoir une séquence d'initialisation (ou de remise à zéro dans le cas d'un compteur). Après l'action, on établit une séquence de décrémentation du décompteur suivi d'une reprise de séquence en fonction de la valeur de celui-ci

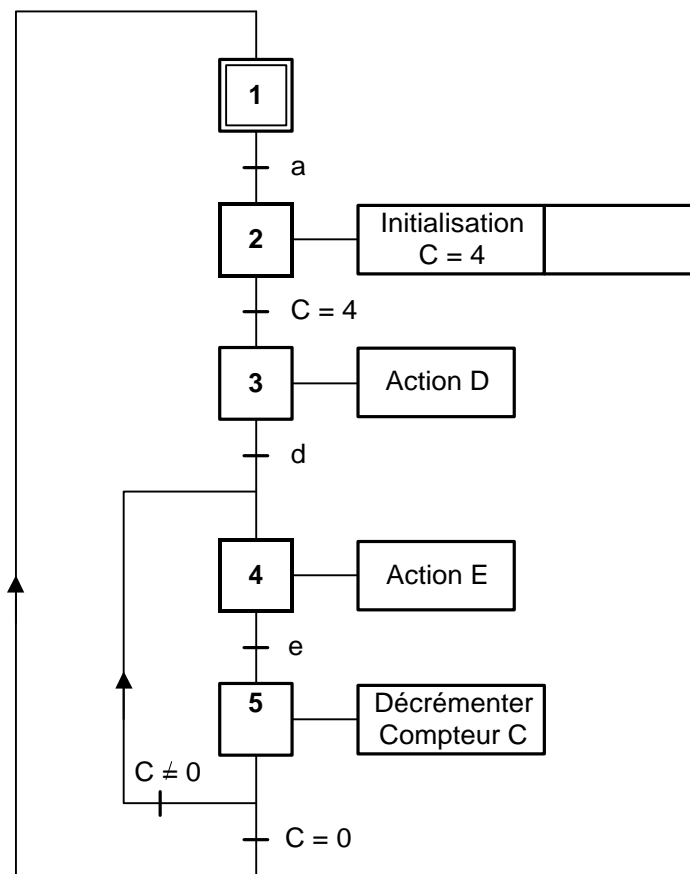


Figure 10 : Décompteur

2.4 Transitions

2.4.1 Transition

Les transitions indiquent les possibilités d'évolution entre étapes. On associe à chaque transition une condition logique appelée réceptivité.

2.4.2 Réceptivité

La réceptivité est écrite sous forme de proposition logique, c'est une information simple ou une fonction combinatoire d'informations extérieures (capteur, compteur, etc ...)

2.4.3 Liaison orientée

Les liaisons indiquent l'évolution de l'état du grafcet.
Les liaisons sont horizontales ou verticales.

2.5 Séquences de base

2.5.1 Grafcet linéaire (séquence unique)

Le Grafcet linéaire ci-dessous, représente un cycle fonctionnel d'une perceuse, c'est une succession d'étape et de transitions.

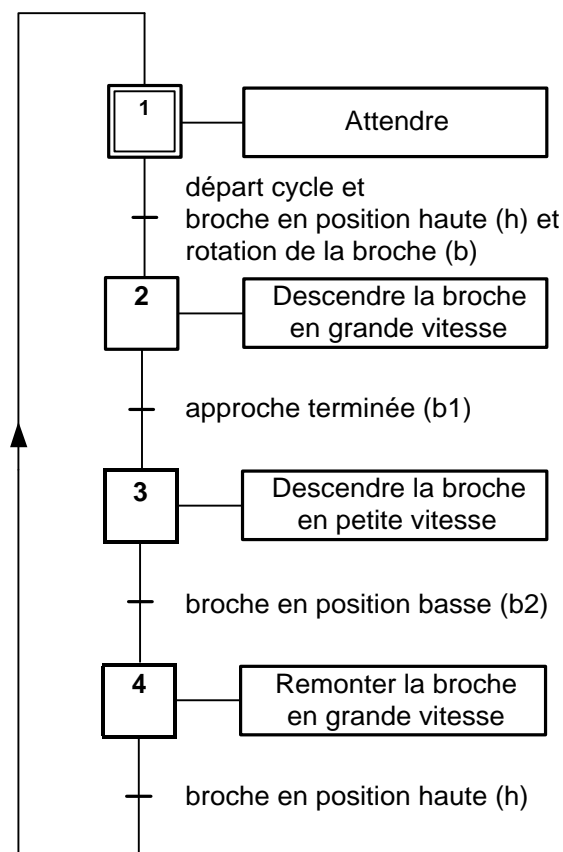


Figure 11 : Grafcet linéaire

2.5.2 Divergence et convergence en "OU".

Choix conditionnel entre plusieurs séquences.

Un GRAFCET est généralement constitué de plusieurs séquences, c'est-à-dire de plusieurs suites d'étapes à exécuter les unes après les autres et il est souvent nécessaire d'effectuer une sélection exclusive d'une de ces séquences.

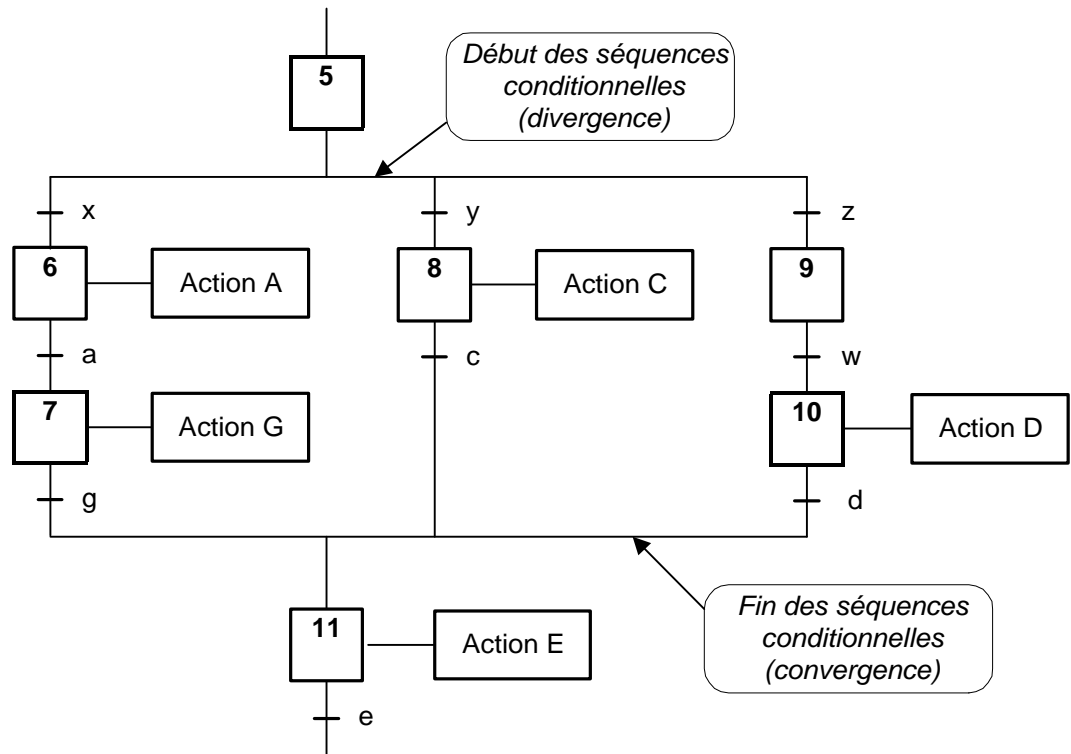
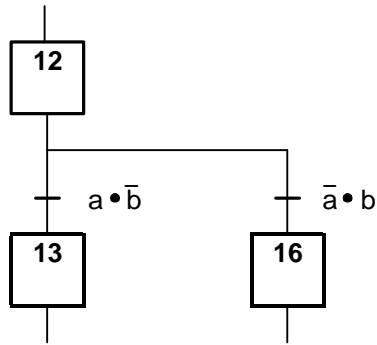


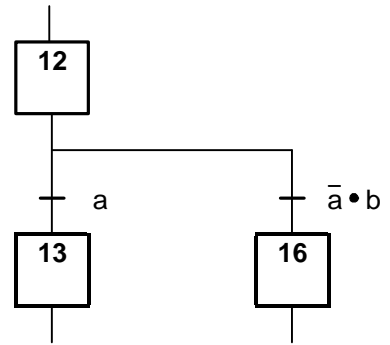
Figure 12 : Divergence et convergence en "OU"

RM : Une divergence entraîne automatiquement une convergence.

Dans l'aiguillage formé par le choix de la séquence à réaliser, les différentes transitions correspondant aux réceptivités x, y et z étant simultanément validées par la même étape 5, pourraient, d'après la règle 4 de simultanéité, être franchies simultanément. En pratique, l'automaticien est souvent amené à rendre des réceptivités exclusives. Il est possible également d'introduire des priorités.



Réceptivités $a\bar{b}$ et $\bar{a}b$ exclusives
 Si a et b sont présents à la fois,
 aucune transition ne pourra être
 franchie à partir de l'étape 12.



Priorité à la réceptivité a . La
 priorité donnée à la transition
 12-13 permet à celle-ci d'être
 franchie lorsque a et b sont
 présents en même temps.

Figure 13 : réceptivités exclusives.

2.5.3 Divergence et convergence en "ET"

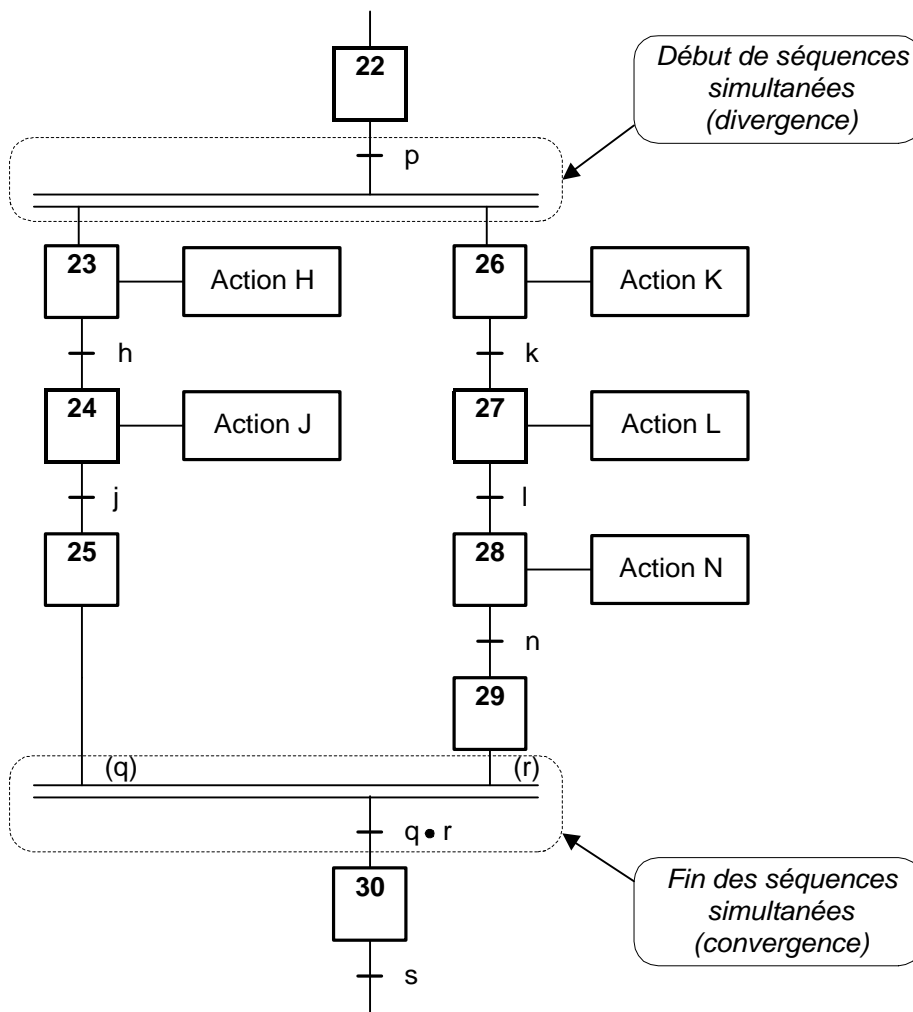


Figure 14 : divergence et convergence en ET.

Un GRAFCET peut comporter plusieurs séquences s'exécutant simultanément mais dont les évolutions des étapes actives dans chaque branche restent indépendantes.

Pour représenter ces fonctionnements simultanés, une transition UNIQUE et deux traits parallèles indiquent le début et la fin des séquences, c'est-à-dire l'activation simultanée des branches ainsi réalisées et leur attente réciproque vers une séquence commune

A partir de l'étape 22 de la figure 14, la réceptivité p provoque l'activation simultanée des étapes 23 et 26.

Ces deux séquences 23-24-25 et 26-27-28-29 évolueront alors de façon totalement indépendante et ce n'est que :

- lorsque les étapes de fin de branche 25 et 29 sont actives,
 - lorsque la réceptivité est vraie ($q \cdot r = 1$),
- que la transition sera franchie. L'étape 30 devient alors active et les étapes 25 et 29 inactives.

Nota : Les conditions particulières à chaque branche peuvent être notées entre parenthèses au-dessus des traits parallèles de regroupement.

2.5.4 Sauts d'étapes et reprise de séquences

Le saut conditionnel est un aiguillage particulier permettant de sauter une ou plusieurs étapes lorsque les actions à réaliser deviennent utiles, tandis que la reprise de séquences permet au contraire de reprendre une ou plusieurs fois la même séquence tant qu'une condition fixée n'est pas obtenue.

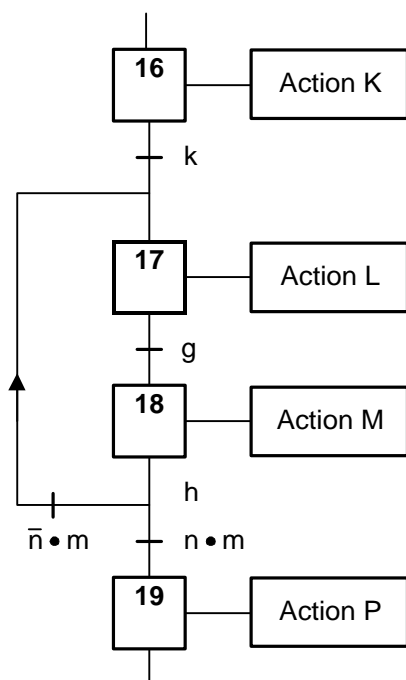


Figure 15 : saut d'étapes

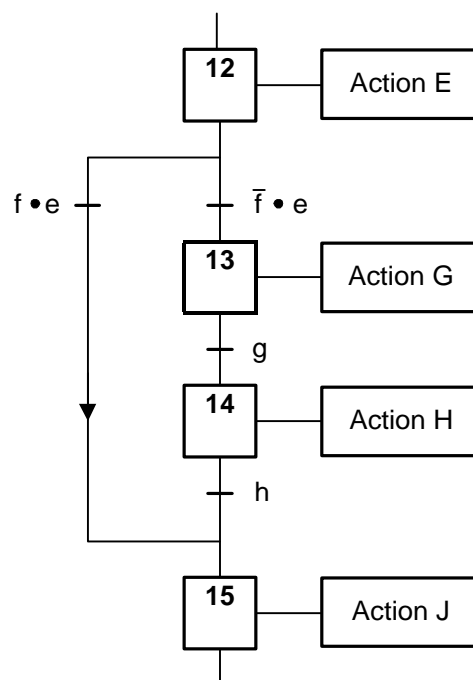


Figure 16 : reprise de séquence

2.5.5 Sous-programmes

Le besoin de sous-programmes en informatique et en automatique relève de deux objectifs différents :

- un objectif de structuration, c'est-à-dire de programmation structurée, particulièrement visé dans les langages comme PASCAL ET ADA. On aboutit alors à une structure de GRAFCET multigraphes dont la réalisation en technologie programmée pourra utiliser des concepts informatiques de sous-programmes ou procédures,
- un objectif d'économie d'écriture, (c'est-à-dire de minimisation de lignes du programme, sans aucun rapport avec l'application, dans le cas d'une réalisation programmée). A cet objectif correspond le cas des automatismes comportant des tâches répétitives. Le GRAFCET permet évidemment lui aussi de décrire ces simplifications d'écritures.

a) Macro-étape

Une macro-étape, répétée autant de fois que nécessaire, est introduite dans le graphe principal à la place de la séquence répétitive. Cette séquence est décrite au voisinage immédiat du graphe. La macro-représentation répond aux besoins d'approche progressive et structurée des applications industrielles complexes.

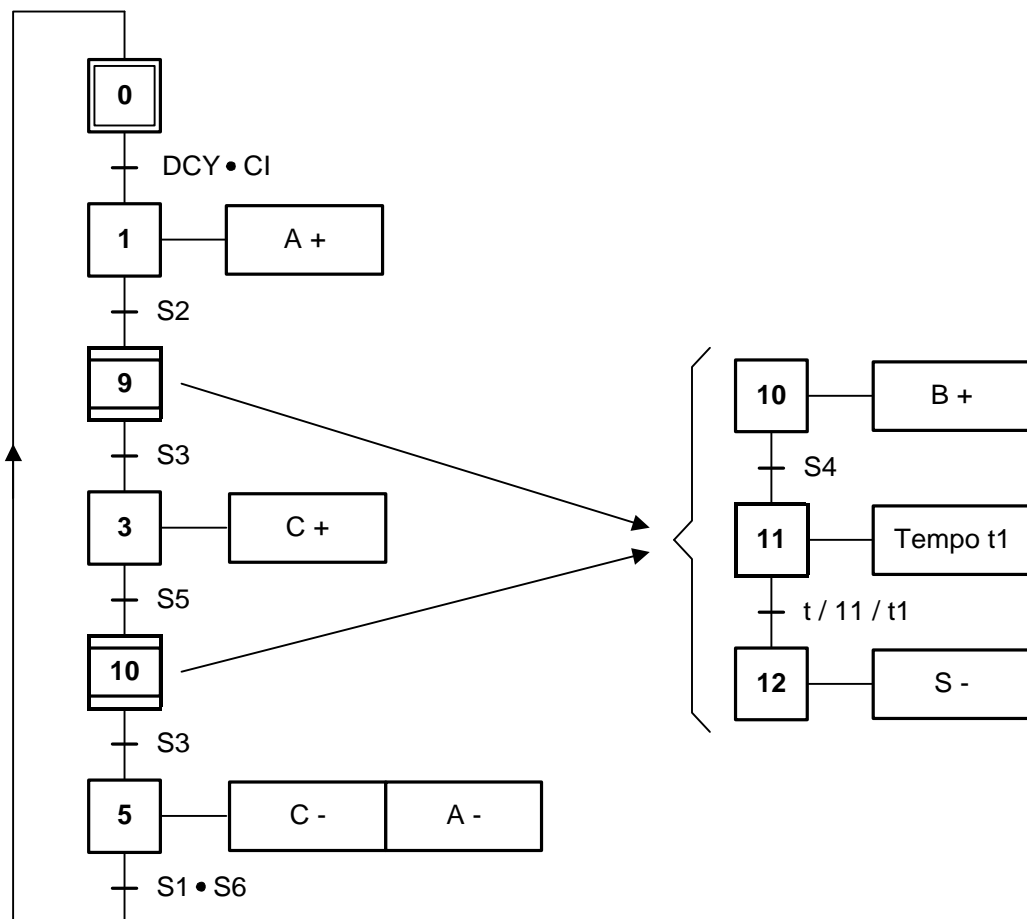


Figure 17 : macro étape

Par rapport à la réalisation programmée, l'étape 10 correspond au saut dans le sous-programme (appel du sous-programme) et son activité est matérialisée par le pointeur programme principal.

Cependant, la programmation ne sera pas exactement conforme à cette description car, en général, le test de S3 est incorporé dans le sous-programme.

b) Graphe auxiliaire

Les étapes 2 et 4 sont des étapes d'appel du graphe de séquence répétitive et provoquent l'activation de l'étape 10 par franchissement de la transition (1), réceptive à l'activité de ces étapes (X2 ou X4). L'étape 13 est l'étape de retour au graphe principal.

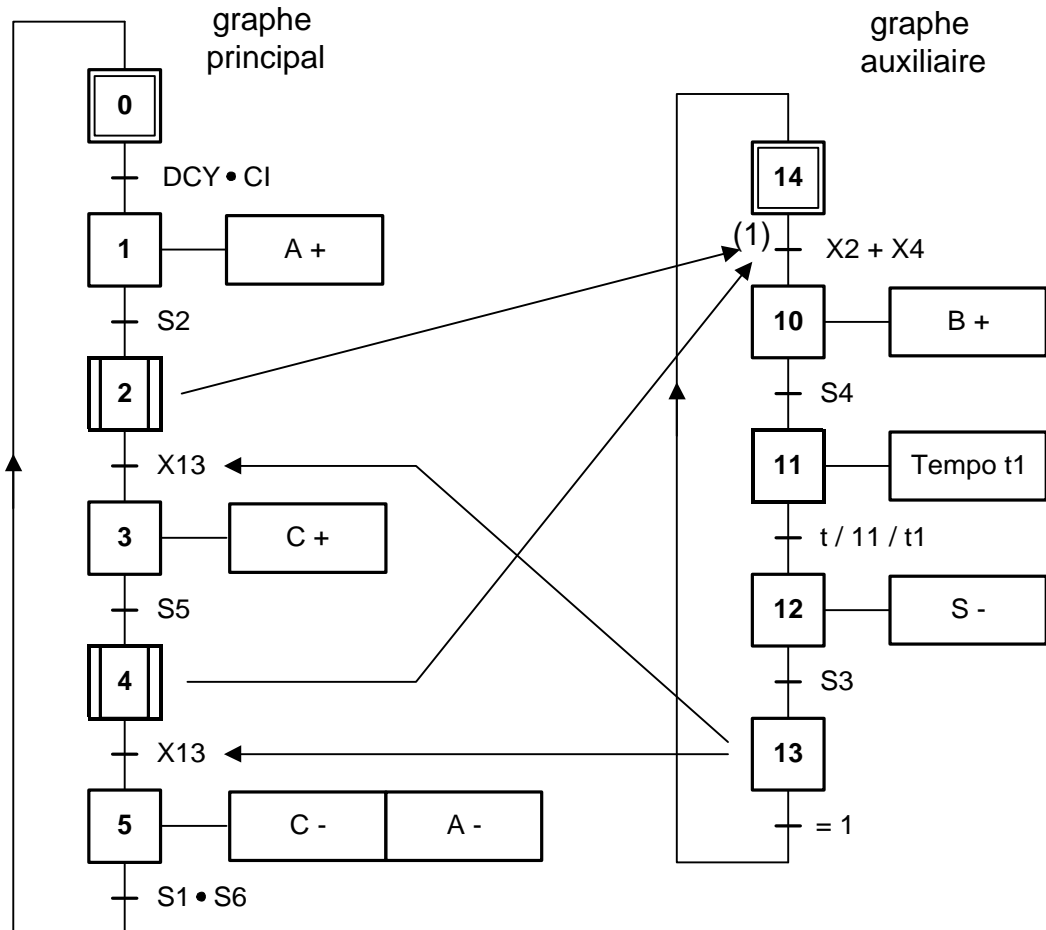


Figure 18 : graphe auxiliaire

Il existe d'autres descriptions fonctionnelles possibles : exemple utilisant les concepts de transitions source et puits avec franchissements simultanés de transitions (*).

En conclusion, il existe de multiples représentations fonctionnelles possibles pour décrire l'utilisation d'une même ressource dans une séquence, à des instants différents.

Le problème peut être beaucoup plus complexe dans certains cas (appel de sous-programmes paramétrés ou multiplexés, la gestion d'appels multiples, etc ...)

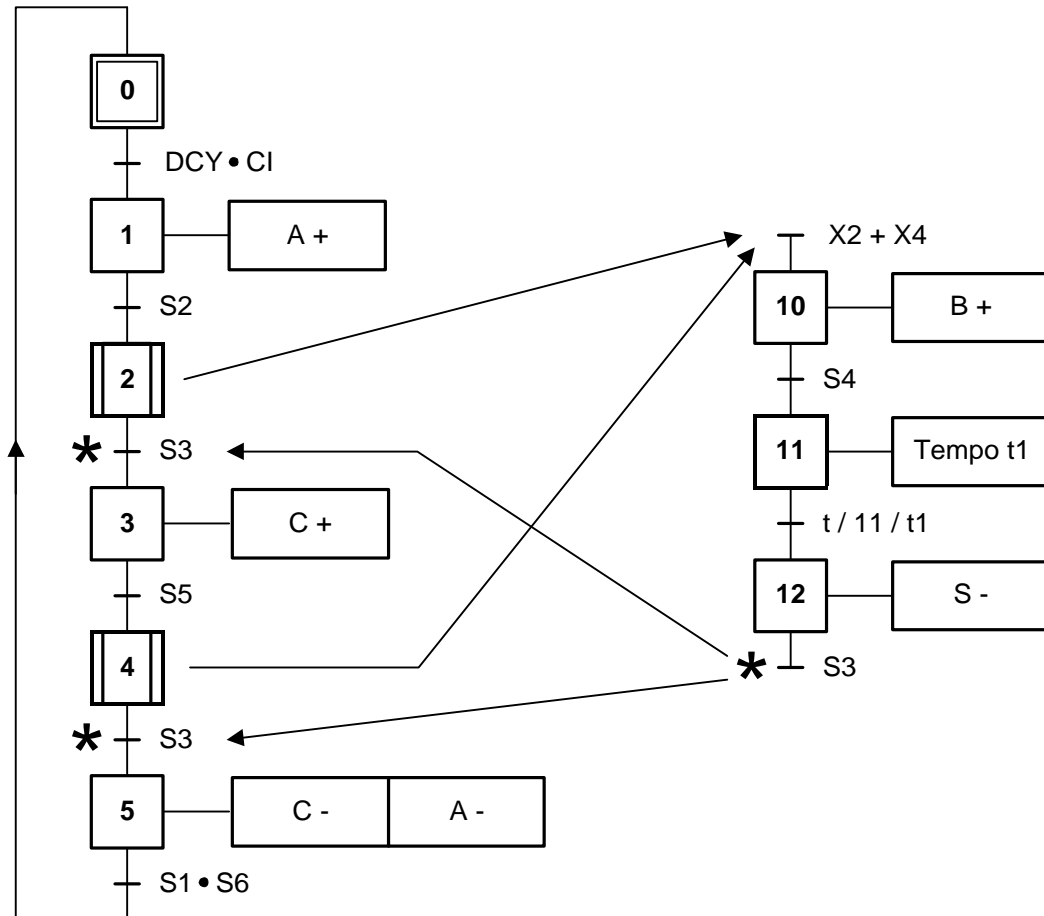


Figure 19 : graphe auxiliaire

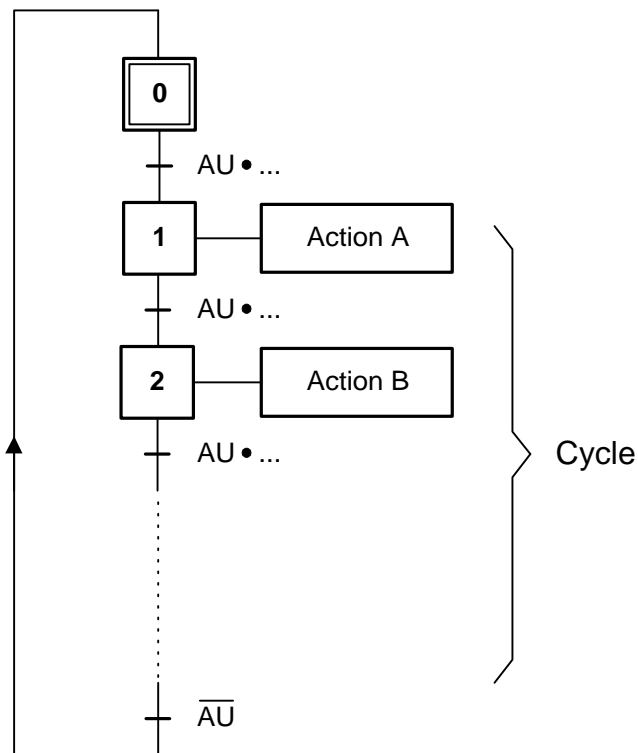
2.5.6 Traitement d'arrêt

Deux solutions peuvent être proposées :

- **sans séquence d'urgence** : le système arrête son évolution,
- **avec séquence d'urgence** : l'apparition "AU" déclenche une procédure particulière.

a) sans séquence d'urgence "Figé"

Figeage de l'automatisme : on suppose dans ce cas que le terme "AU" est un facteur implicite dans toutes les réceptivités. Dès l'occurrence de "AU", aucune réceptivité ne peut prendre la valeur 1 : la situation ne peut évoluer. La disparition de "AU" autorise à nouveau l'évolution du cycle.



CI : conditions
initiales

Figure 20 : traitement d'arrêt figé.

b) séquence d'urgence

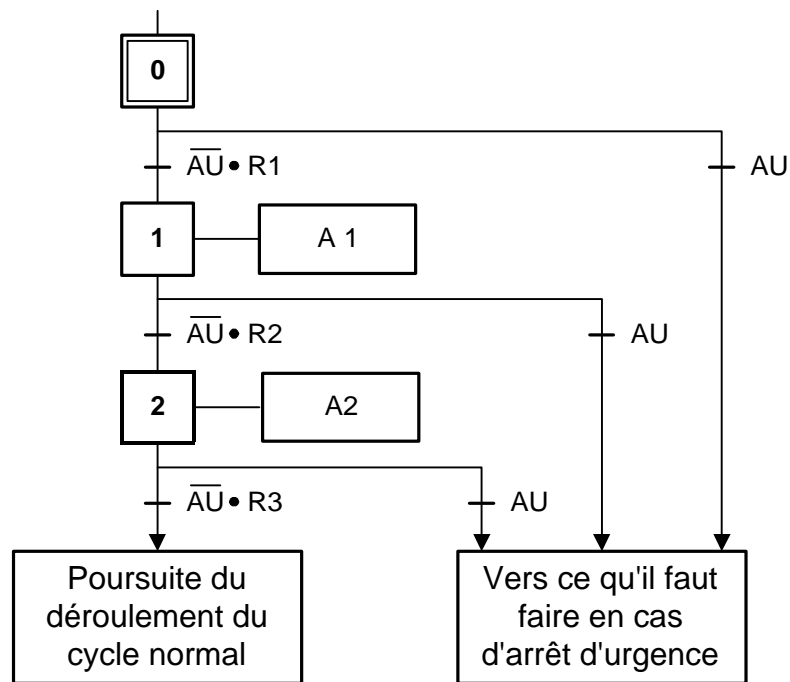


Figure 21 : traitement d'arrêt d'urgence séquentiel.

On sait qu'un des intérêts du GRAFCET est de n'indiquer à chaque situation que les seules circonstances qui la font évoluer.

L'ARRET D'URGENCE devant toujours faire évoluer la situation de la partie commande, on devrait, en toute rigueur, la faire apparaître à chaque étape du GRAFCET.

Dès que le cas traité devient un peu important, le GRAFCET devient complètement illisible. Il en découle qu'il est malaisé de considérer l'arrêt d'urgence comme une information d'entrée comme les autres. De plus, c'est contraire à l'esprit même de l'urgence accordant à cet arrêt une sorte de "super priorité". Pourtant il est intéressant (mais ce n'est pas toujours possible) de pouvoir formaliser le comportement attendu d'un système lors de l'apparition de l'information "AU."

2.5.7 Traitement de modes de marche

a) Marche cycle par cycle

Chaque cycle se déroule automatiquement mais nécessite chaque fois une nouvelle intervention de l'opérateur sur "DCY" pour pouvoir exécuter le cycle suivant.

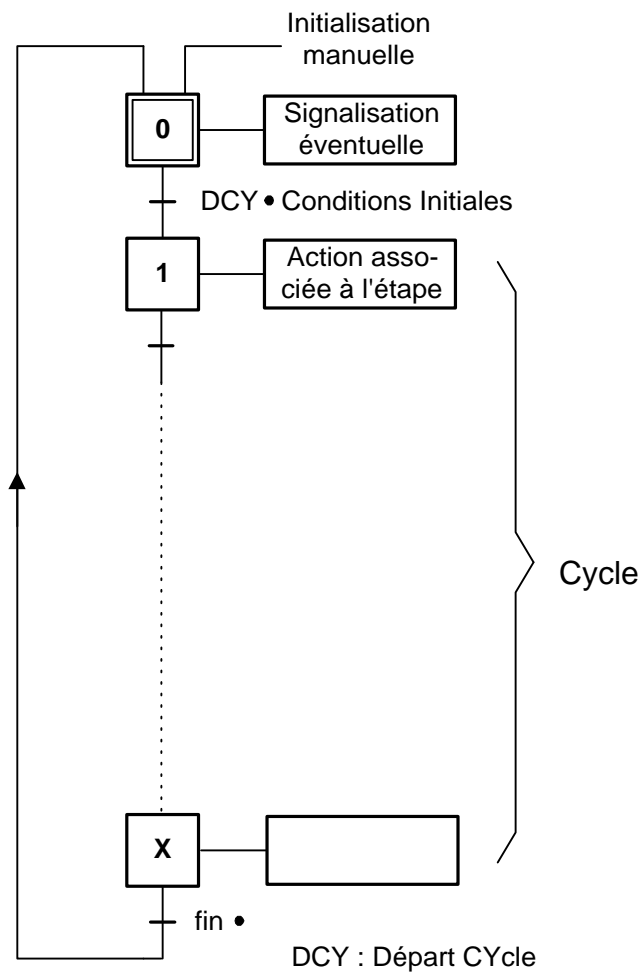


Figure 22 : Marche cycle par cycle

b) Cycle unique

Le cycle ne se déroule qu'une fois, même si l'information "Départ cycle" est maintenue présente par l'opérateur.

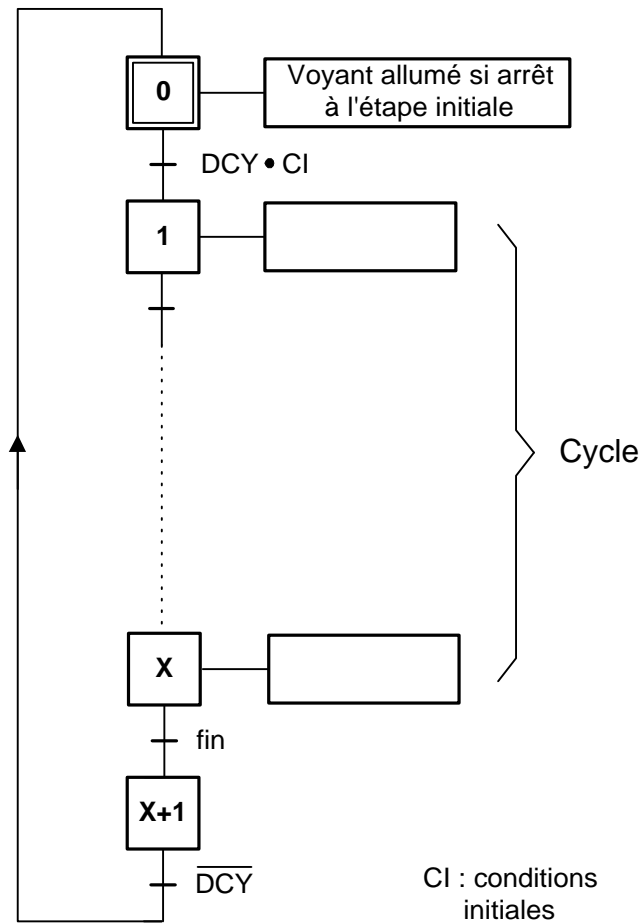


Figure 23 : Marche cycle unique

c) **Marche automatique / arrêt cycle**

Après une action sur "DCY", le cycle se répète indéfiniment jusqu'à ce que l'ordre d'arrêt soit donné, cet arrêt ne s'effectuant qu'une fois le cycle terminé.

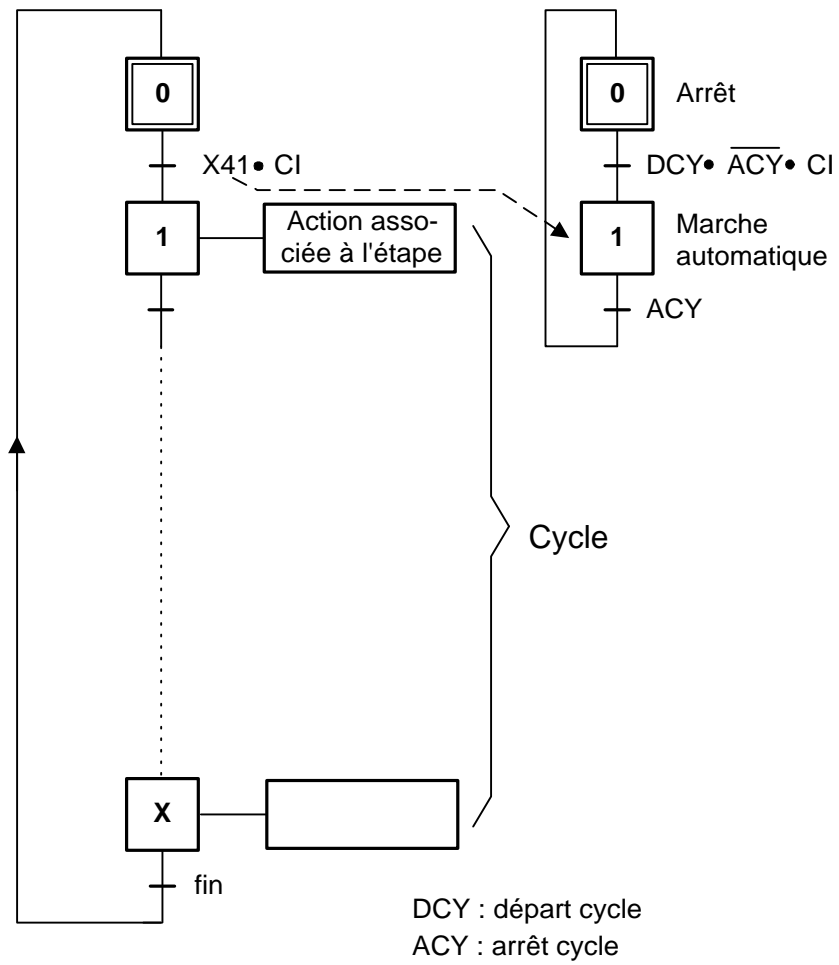


Figure 24 : Marche automatique / arrêt cycle

d) Marche auto/cycle par cycle

Le cycle démarre par une action sur "DCY" et c'est ensuite le rebouclage du cycle en fonction de la position du commutateur.

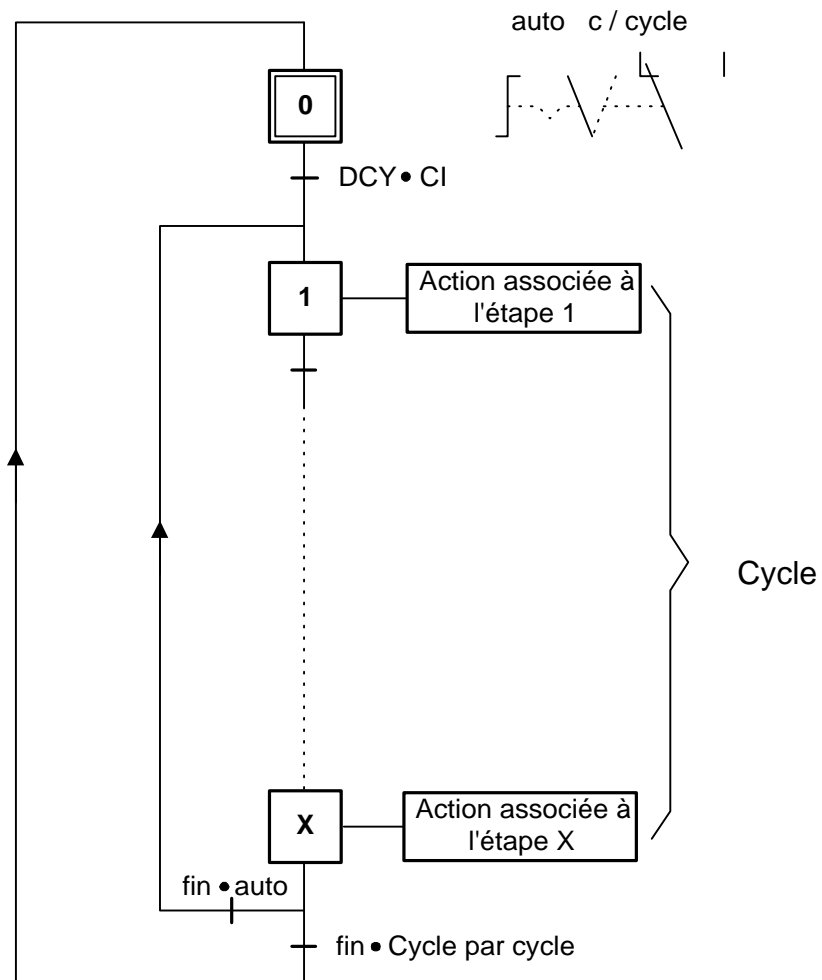


Figure 25 : Marche auto / cycle par cycle

Réalisation du Grafcet par câblage

Le but de ce chapitre est de vous montrer comment mettre en oeuvre un Grafcet à l'aide de composants d'électronique ToR (portes et bascules).

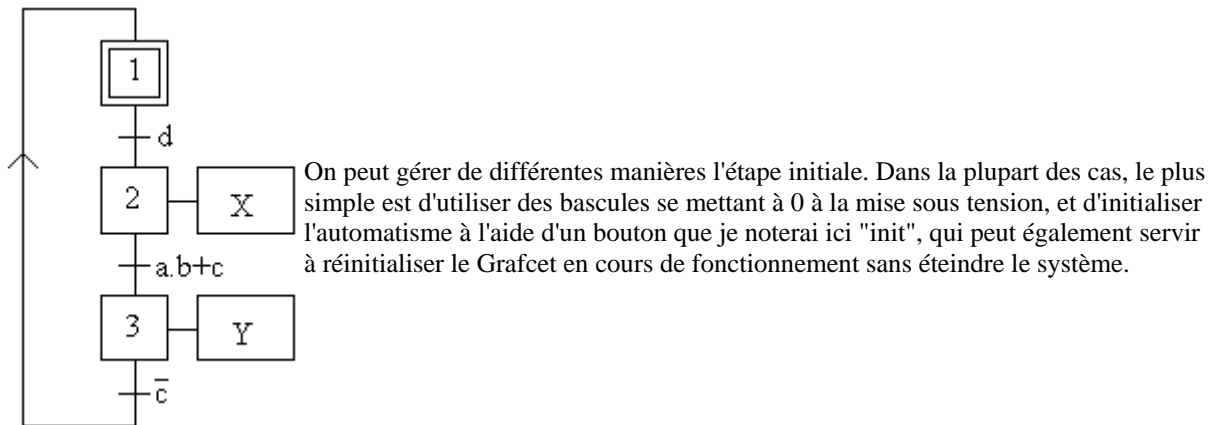
1 Cas sans problèmes

Nous allons d'abord voir les cas simples, par une méthode qui ne vérifie pas intégralement toutes les règles du Grafcet. Si j'en parle, c'est parce que les cas nécessitant plus de précautions sont rares et faciles à identifier.

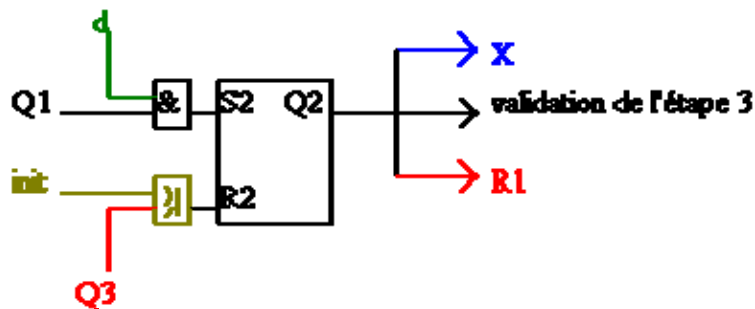
1.1 Grafcet linéaire

Il suffit d'utiliser une bascule RS par étape. Une étape est allumée si l'étape précédente est active et que la réceptivité d'entrée est vraie. Dans le cas d'un Grafcet linéaire, on désactivera une étape quand la suivante est active. Ceci simplifie le câblage, mais ne respecte pas toutes les règles du Grafcet (en fait cette méthode fonctionne dans une très grande majorité de cas, nous traiterons les cas litigieux plus loin dans ce document).

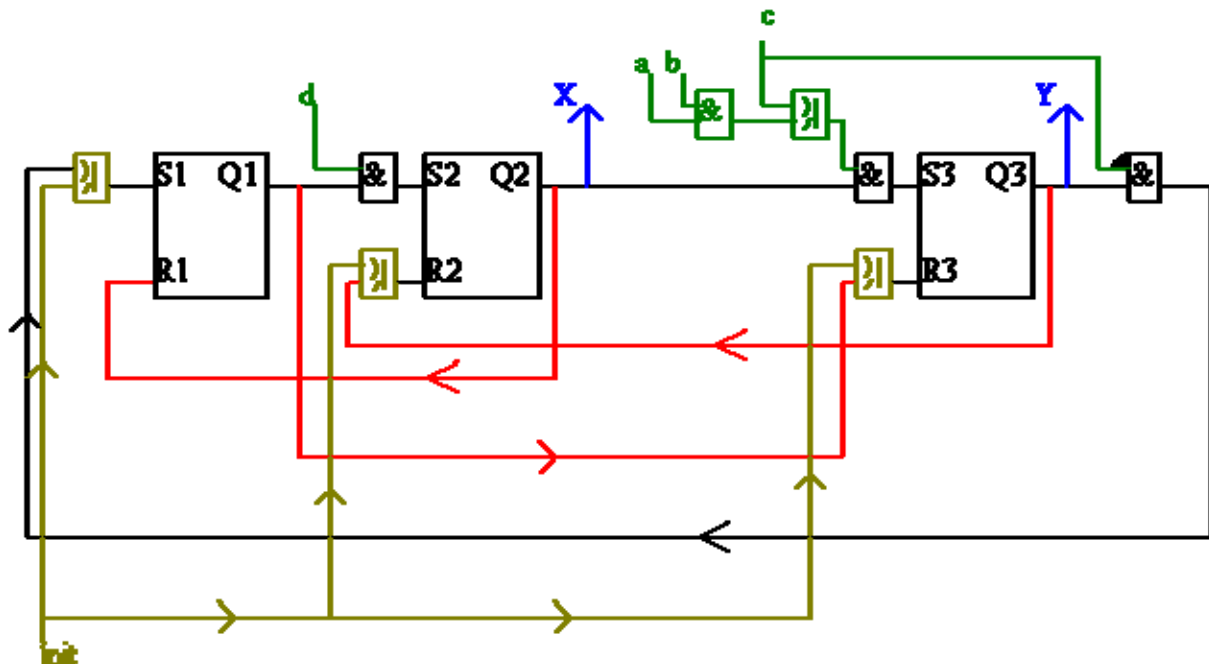
Soit le Grafcet :



Notons, pour l'étape numéro i , son entrée Set par S_i , son entrée Reset par R_i , sa sortie Q_i . Etudions l'étape 2. Elle s'allume si l'étape 1 est active et d est vrai ($S_2=Q_1.d$). Tout le temps quelle est active, la sortie X est allumée ($X=Q_2$). Elle s'éteint normalement quand la réceptivité de sortie est vraie, mais (comme précisé plus haut) nous allons attendre pour éteindre l'étape 2 que l'étape 3 soit active (donc $R_2=Q_3$), et donc être sûr que l'étape 3 a eu le temps de prendre en compte l'information. Elle peut également être éteinte par init, puisqu'elle n'est pas initiale.



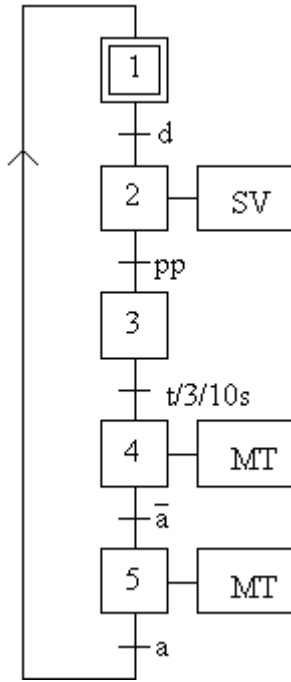
Il suffit de répéter cela pour chaque étape et relier le tout. Le schéma de câblage du système complet sera donc (j'ai gardé la même disposition que le Grafcet, mais retourné de 90 degrés, les électroniciens préfèrent les entrées à gauche et les sorties à droite) :



L'étude de chaque étape est simple, la principale difficulté est le routage (c'est à dire relier le tout), surtout si l'on veut faire un circuit imprimé (où les croisements de pistes sont impossibles). D'autant plus que chaque composant doit être alimenté, mais je n'ai pas représenté ici les alimentations. Mais il existe désormais de bons logiciels de routage.

On peut déjà conclure que si la mise en oeuvre d'un Grafcet par câblage n'est pas très compliquée, la modification est pour le moins difficile. En général, on préférera refaire un nouveau câblage si l'on désire modifier le Grafcet. De même, le câblage a intérêt à être complètement testé dès sa réalisation, la recherche d'erreurs après coup étant bien plus difficile.

Exercice : câbler ce Grafcet de 5 étapes gérant une amenée de pièces :

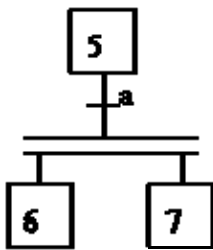


Cahier des Charges :

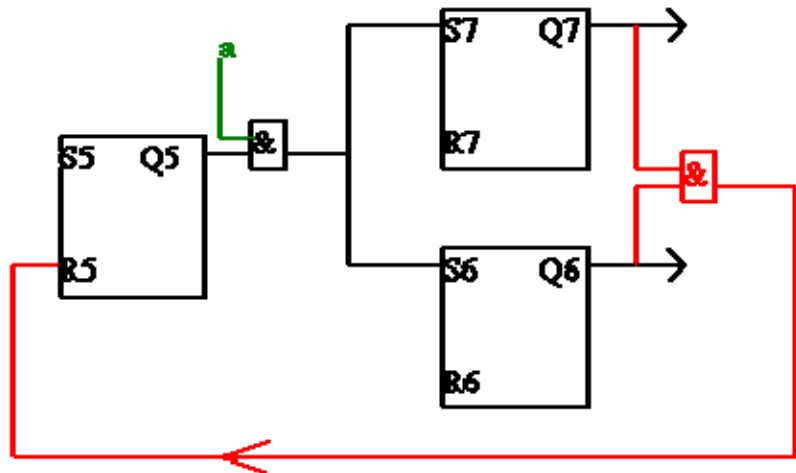
à l'appui de d (départ), on actionne un vérin monostable par l'action SV, jusqu'à ce que la pièce soit positionnée sur le tapis. Puis on attend 10 secondes (retour du vérin), puis on enclenche le moteur du tapis roulant (MT) jusqu'à ce que la pièce arrive sur le capteur a. Comme la pièce précédente était peut-être en a au début du cycle, il faut attendre un front montant de a, que je gère en attendant que a soit d'abord relâché puis à nouveau appuyé. La temporisation sera réalisée par un composant réglable (en fait un circuit RC avec une résistance variable), qui donne 1 à sa sortie si son entrée est à 1 pendant au moins le temps requis.

1.2 Divergence simple en ET

Quand la transition est franchissable, il suffit d'allumer deux étapes au lieu d'une. Le seul problème est la désactivation de l'étape précédente : il faut être sûr que les deux étapes suivantes ont eu le temps de prendre en compte l'information d'activation avant de désactiver la précédente (si l'on désactive dès qu'une des deux est active, la seconde ne s'activera plus).



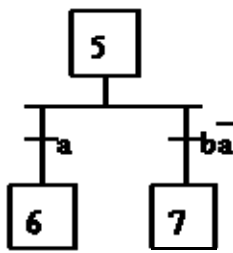
je ne traite ici ni l'amont, ni l'aval, ni les actions, uniquement les liaisons entre 5 et ses suivantes.



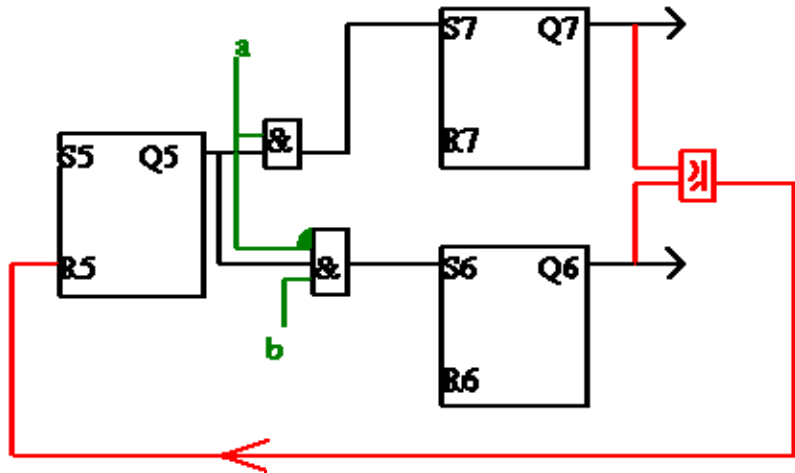
Ce câblage simple ne répond pas aux règles du Grafset si 5 peut être réactivé avant que 6 et 7 n'aient été désactivées. Il en est de même si l'étape 7 par exemple peut être activée d'une autre manière (convergence en OU). Ces cas sont cependant très rares dans la pratique.

1.3 Divergence exclusive en OU

Il n'y a aucun problème particulier.



Comme au dessus, je ne traite ici que les liaisons entre 5 et ses suivantes.

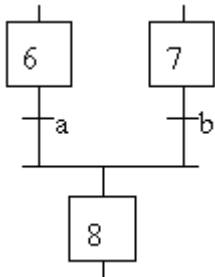


Si la divergence n'est pas exclusive (les deux réceptivités peuvent être vraies en même temps), c'est un peu plus compliqué, le mieux est de traiter les trois cas (l'une seule, l'autre seule, les deux).

1.4 Convergence en ET

Je ne fais pas le schéma, il est évident : il faut que les (deux en général) étapes précédentes soient actives, et la réceptivité vraie, pour activer l'étape suivante, celle ci désactivant les étapes précédentes.

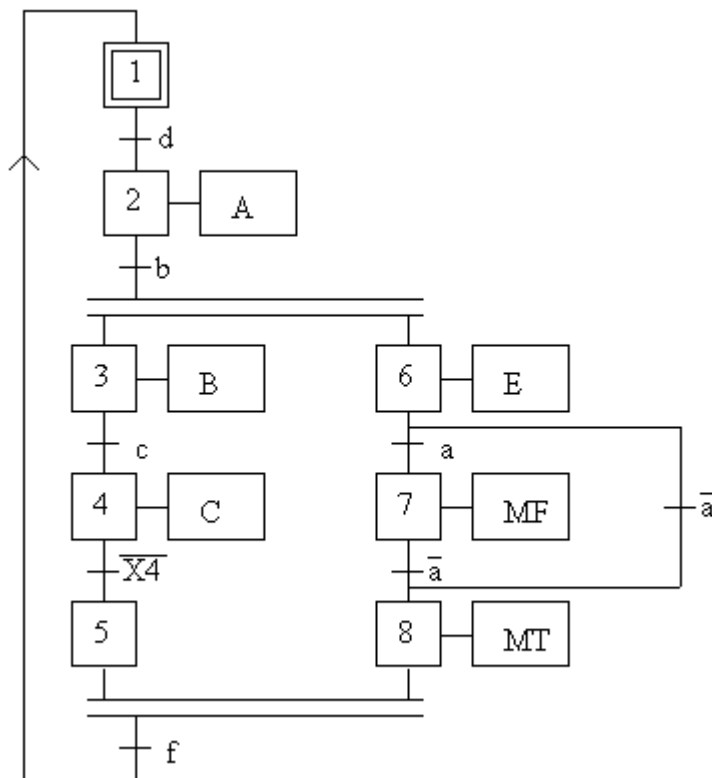
1.5 Convergence simple en OU



Vu le temps que je mets pour faire un schéma (le seul outil dont je dispose est paintbrush, et comme je suis en vacances je ne dispose que d'un portable à écran monochrome, nom parfaitement choisi puisqu'il n'est même pas noir et blanc mais gris et gris), je me contente de l'expliquer (ça vous fera un bon exercice).

On allume 8 si (6 et a) ou (7 et b). On éteint 6 et 7 tant que l'on a 8. Evidement ceci ne fonctionne que si l'on ne peut pas avoir simultanément 6 et 7 actives, mais j'ai bien dit (dans le titre ci-dessus) que je ne traite que le cas simple, qui de plus se trouve être aussi le plus courant.

1.6 Exercice récapitulatif



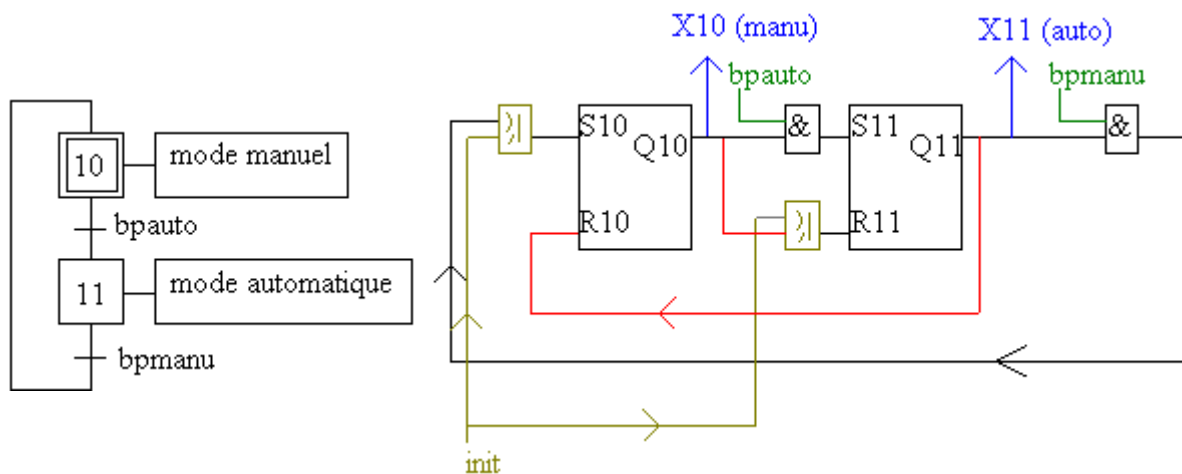
Câblez ce Grafcet (il ne pose pas de problème particulier). Ce Grafcet regroupe les différents cas de divergence - convergence.

cliquez [ici](#) pour la solution

2 Cas où cette méthode est mauvaise

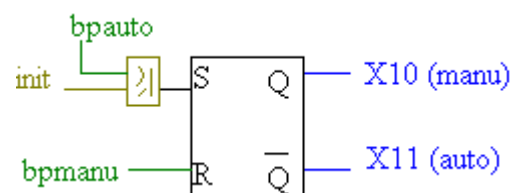
2.1 Grafcet à deux étapes

Soit le Grafcet suivant, et sa réalisation d'après la méthode précédente :



Quand 10 est actif (Q10) et bpauto vrai, en même temps on essaie d'allumer 11 par S11 et de l'éteindre par R11. Même en prenant une bascule à priorité déclenchement, l'état de 11 sera celui du dernier signal sur ses broches, ce qui risque d'être aléatoire.

Ici, la solution est simple : une seule bascule suffit. Mais cet exemple montre bien le problème de ces câblages : une étape désactive la précédente tant qu'elle même est active, au lieu de ne le faire qu'au moment du franchissement de la transition.

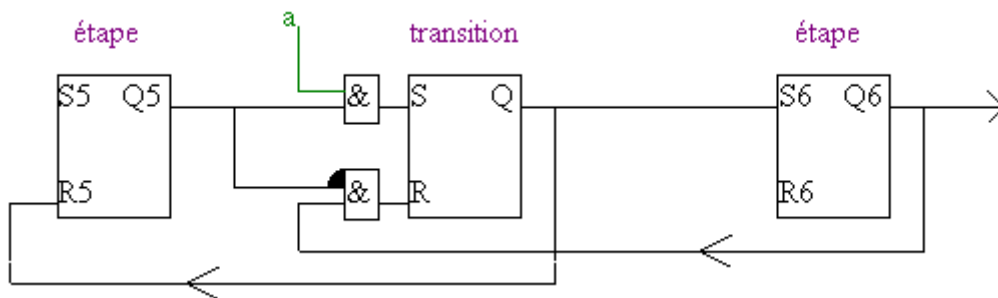


Le problème vient de la désactivation. Tous les composants ne peuvent pas avoir tous exactement un même temps de réponse. Donc puisqu'on active une étape quand la précédente est active et la réceptivité est vraie, si simultanément on désactivait la précédente il est possible que la suivante n'ait pas eu le temps de s'activer avant que le signal ne disparaisse. La solution choisie est sûre, mais l'information de désactivation est envoyée bien plus longtemps que nécessaire. Pour être sûr du résultat il faudrait mémoriser (dans une bascule) l'état de chaque transition. En réalisation électronique ce n'est pas le prix qui poserait problème mais la complication du circuit (déjà assez complexe sans cela). En réalisation pneumatique ou électrique s'ajouterait le prix des composants.

2.2 mémorisation de la transition

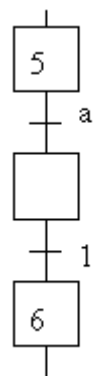
Donc une solution respectant mieux les règles du Grafcet consisterait à utiliser une bascule pour chaque transition. Elle est allumée quand l'étape précédente et la transition sont vraies, sa sortie active l'étape suivante et désactive la précédente. Quand doit on éteindre la bascule représentant la transition ? Le problème reste donc entier. Une bonne solution est de le faire quand le franchissement a été effectué, c'est à dire quand la suivante est active et que la précédente ne l'est pas. Attention, ce cas peut arriver sans que l'on soit passé par cette transition (convergence en OU par exemple), mais dans ce cas on éteint une transition qui l'était déjà, ce qui n'est pas grave.

Faisons donc le schéma de passage entre une étape 5 et une étape 6, reliées par une transition de réceptivité a :



Cette méthode permet de régler le cas où l'étape 5 risque d'être réactivée avant la désactivation de 6.

On peut remarquer que l'on aurait obtenu à peu près le même schéma en modifiant le Grafcet pour qu'il soit compatible avec la première méthode, c'est à dire empêcher qu'il y ait deux étapes successives actives en même temps : il suffit d'intercaler une étape comme représenté ci-contre. C'est une méthode qui permet d'avoir un Grafcet plus proche du câblage, donc un câblage plus clair.

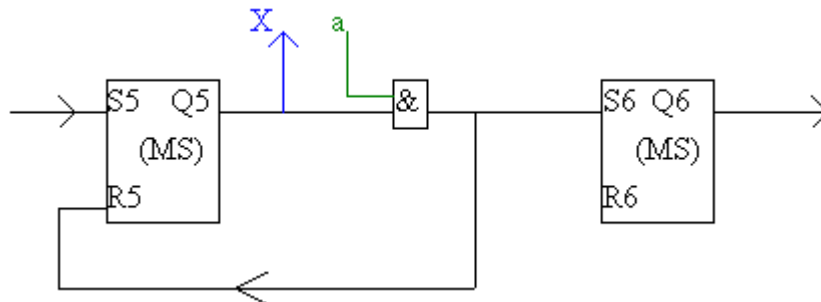


2.3 Bascules synchrones

La méthode précédente peut encore dans certains cas ne pas respecter la règle de simultanéité. Pour cela, une seule solution : synchroniser le fonctionnement des composants. Pour cela, il suffit de prendre la première méthode, mais d'utiliser des bascules MS. Une bascule MS prend en compte les commandes Set et Reset qu'on lui applique non pas immédiatement, mais au prochain front montant de son entrée de synchronisation (horloge). La désactivation d'une étape se fait plus simplement : par la même information que celle qui

active la suivante (les deux seront prises en compte en même temps : au prochain front de l'horloge. Il suffit de choisir une horloge suffisamment rapide pour ne pas ralentir l'automatisme (en général ce point ne pose pas de problème en P.C. électronique), mais plus lente que le temps de réaction du composant le plus lent.

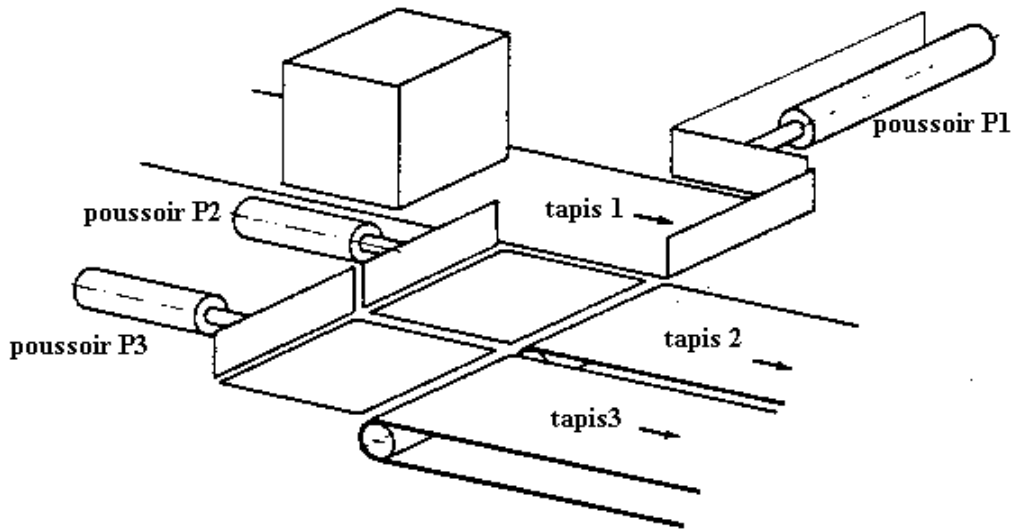
Faisons donc le schéma de passage entre l'étape 5 (d'action X) et l'étape 6, reliées par une transition de réceptivité a :



On peut immédiatement voir que le schéma résultant est grandement simplifié (je n'ai pas représenté l'horloge qui doit être reliée à chaque bascule, comme l'alimentation, plus la gestion de l'initialisation). On peut remarquer qu'une bascule MS est en fait composée de deux bascules RS, et que cette méthode revient à peu près au même que les autres modifications que j'ai proposées (en plus sûr et plus clair). La principale différence est que l'on fixe la durée de l'information de désactivation par un signal d'horloge.

LES EXERCICES

Exercice 1 : Tri de caisses



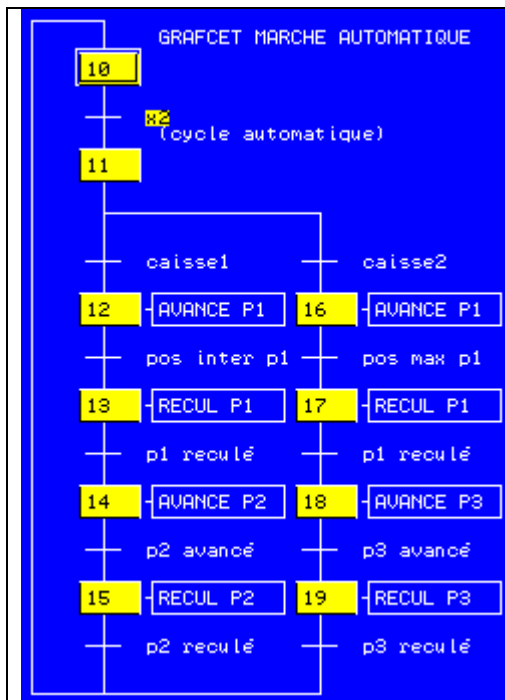
Un dispositif automatique destiné à trier des caisses de deux tailles différentes se compose d'un tapis amenant les caisses, de trois poussoirs et de deux tapis d'évacuation.

Le poussoir P1 pousse les petites caisses devant les poussoir P2 qui à son tour les transfère sur le tapis 2, alors que les grandes caisses sont poussées devant les poussoir P3, ce dernier les évacuant sur le tapis 3.

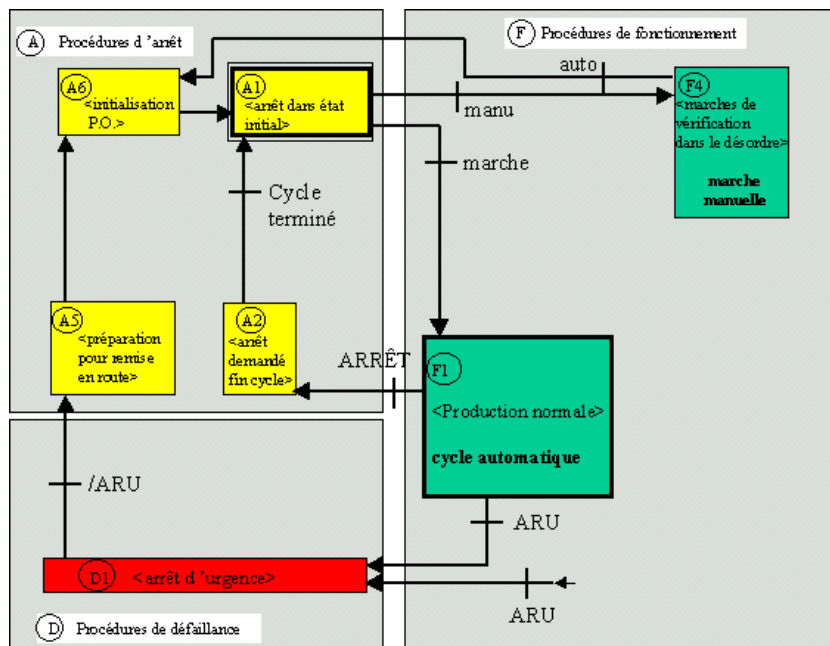
Pour effectuer la sélection des caisses, un dispositif de détection placé devant le poussoir P1 permet de reconnaître sans ambiguïté le type de caisse qui se présente ($a=1$ si petite caisse, $b=1$ si grande caisse).

Représenter le GRAFCET de commande de cette machine.

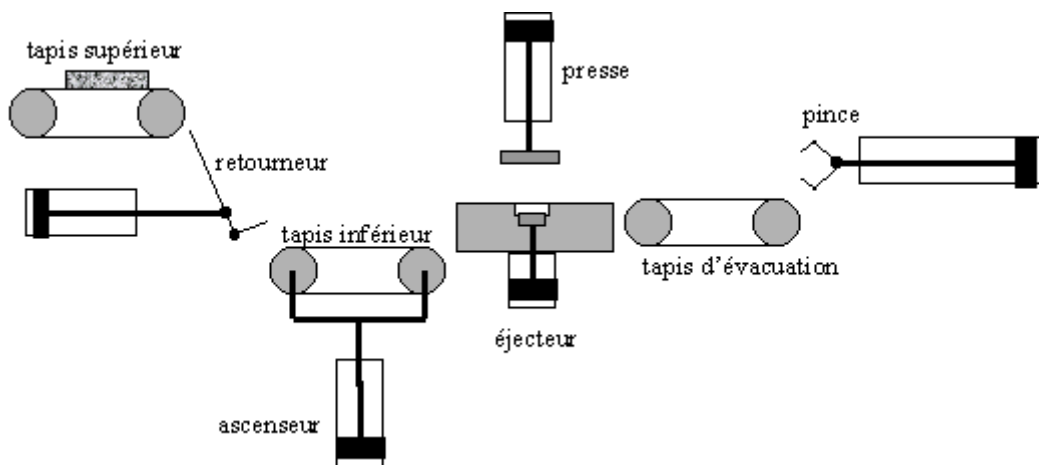
capteurs	actionneurs
pp1 : i1 ;présence pièce a	
pp2 : i2 ;présence pièce b	
v11 : i10 ;poussoir P1 rentré	_v1-_ : o10 ;rentrée poussoir P1
v12 : i11 ; poussoir P1 position intermédiaire	_v1+_ : o11 ;sortie poussoir P1
v13 : i12 ; poussoir P1 sorti	
v21 : i20 ;poussoir P2 rentré	_v2-_ : o20 ;rentrée poussoir P2
v22 : i21 ;poussoir P2 sorti	_v2+_ : o21 ;sortie poussoir P2
v31 : i30 ;poussoir P3 rentré	_v3-_ : o30 ;rentrée poussoir P3
v32 : i31 ;poussoir P3 sorti	_v3+_ : o31 ;sortie poussoir P3



On peut décrire tout d'abord les modes de marche souhaité par un GEMMA:



Exercice 2 : Press object

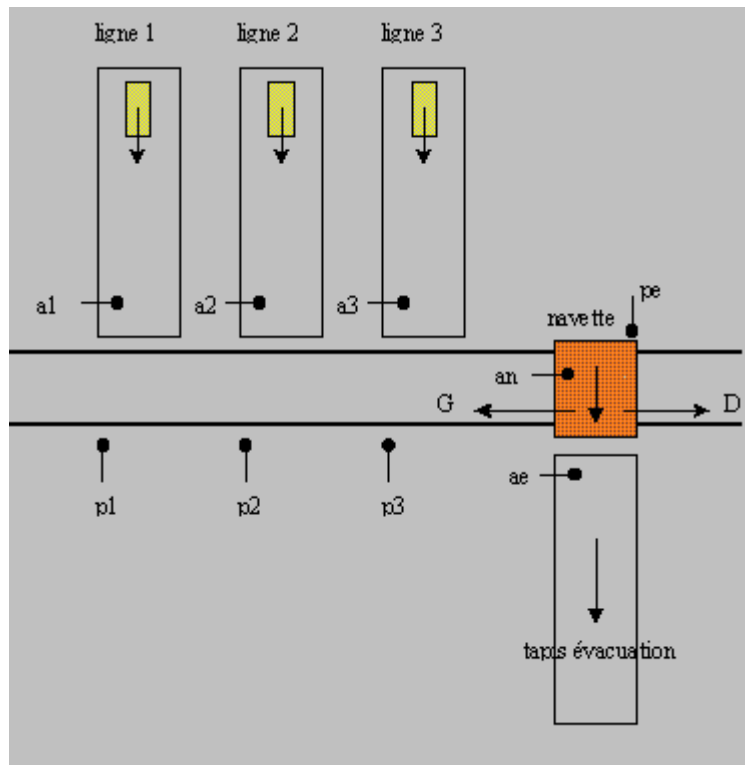


Sensors	Actions
dcy i0 marche	
ppr i2 pièce dans retourneur	_RSUP_ O0 rotation tapis supérieur
rre i1 retourneur reculé	_RER_ O2 recul retourneur
rav i4 retourneur avancé	_AVR_ O1 avance retourneur
ppa i5 présence pièce sur ascenseur	
haut i6 ascenseur en haut	_MAS_ O3 montée ascenseur
bas i9 ascenseur en bas	_DAS_ O5 descente ascenseur
ppp i8 présence pièce sous presse	_RINF_ O4 rotation tapis inférieur

pbas i10 presse en bas	_DPR_ O6 descente presse
phaut i11 presse en haut	_MPR_ O7 montée presse
pav i12 pince avancée	_APINCE_ O8 avance pince
pre i13 pince reculée	_RPINCE_ O9 recul pince

Exercice 3 - priorité

Une navette transporte un par un les produits fabriqués par 3 lignes de fabrication indépendantes vers un tapis roulant d'évacuation selon le schéma ci-dessous :



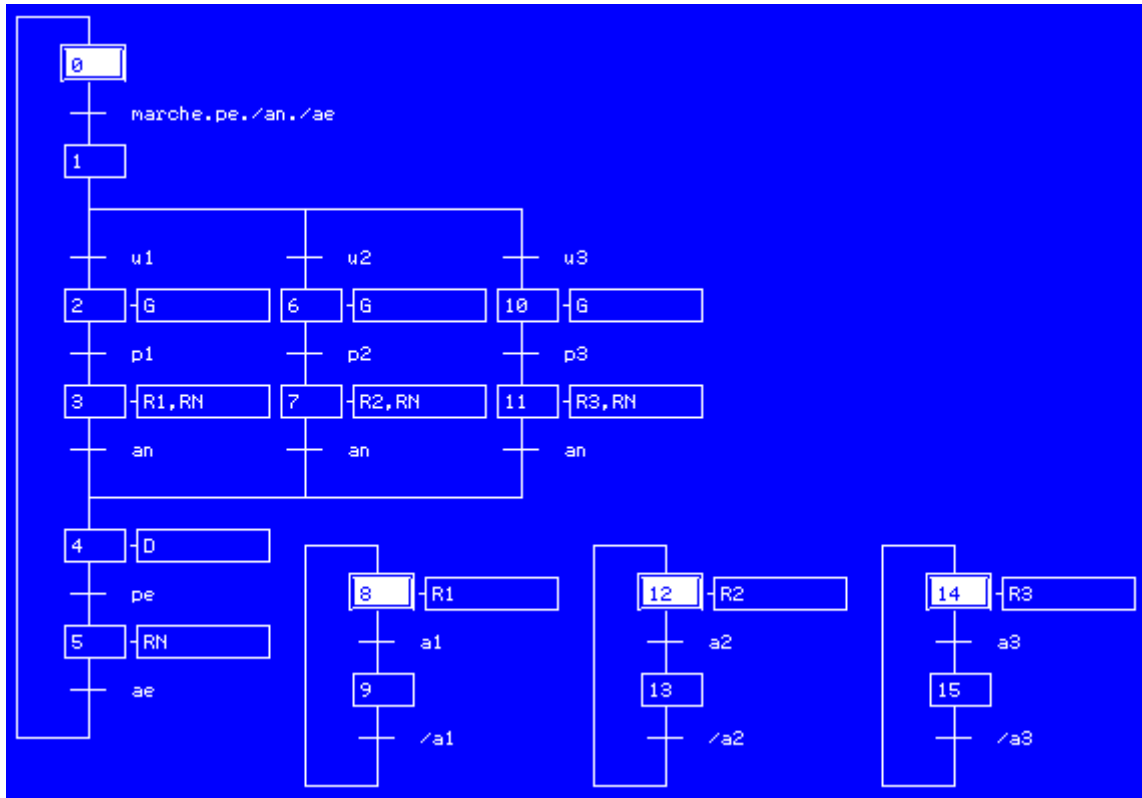
Capteurs	Actions
a1 présence pièce sur ligne 1	G déplacement de la navette vers la gauche
a2 présence pièce sur ligne 2	D déplacement de la navette vers la droite
a3 présence pièce sur ligne 3	R1 rotation tapis ligne 1
an présence pièce sur navette	R2 rotation tapis ligne 2
ae présence pièce sur tapis évacuation	R3 rotation tapis ligne 3
p1, p2, p3, pe détecteurs positions de la navette	Rn rotation tapis navette

en donnant priorité à la ligne la plus éloignée on obtient le GRAFCET ci-dessous dans lequel:

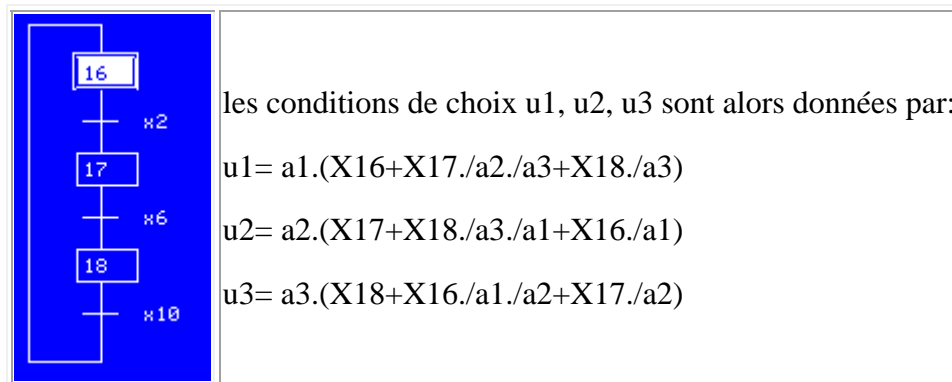
$u1 = a1$

$u2 = /a1.a2$

$u3 = /a1./a2.a3$

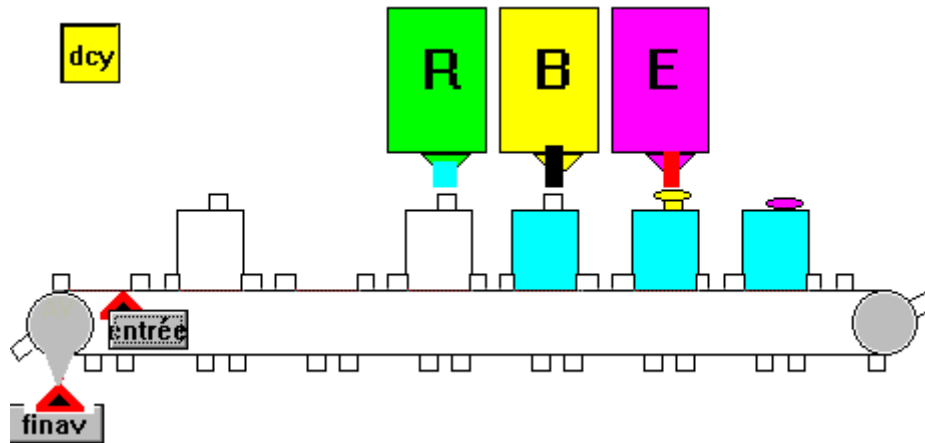


Pour ne pas donner priorité toujours à la même ligne en cas de conflit on peut faire tourner les priorités en ajoutant un GRAFCET des priorités:

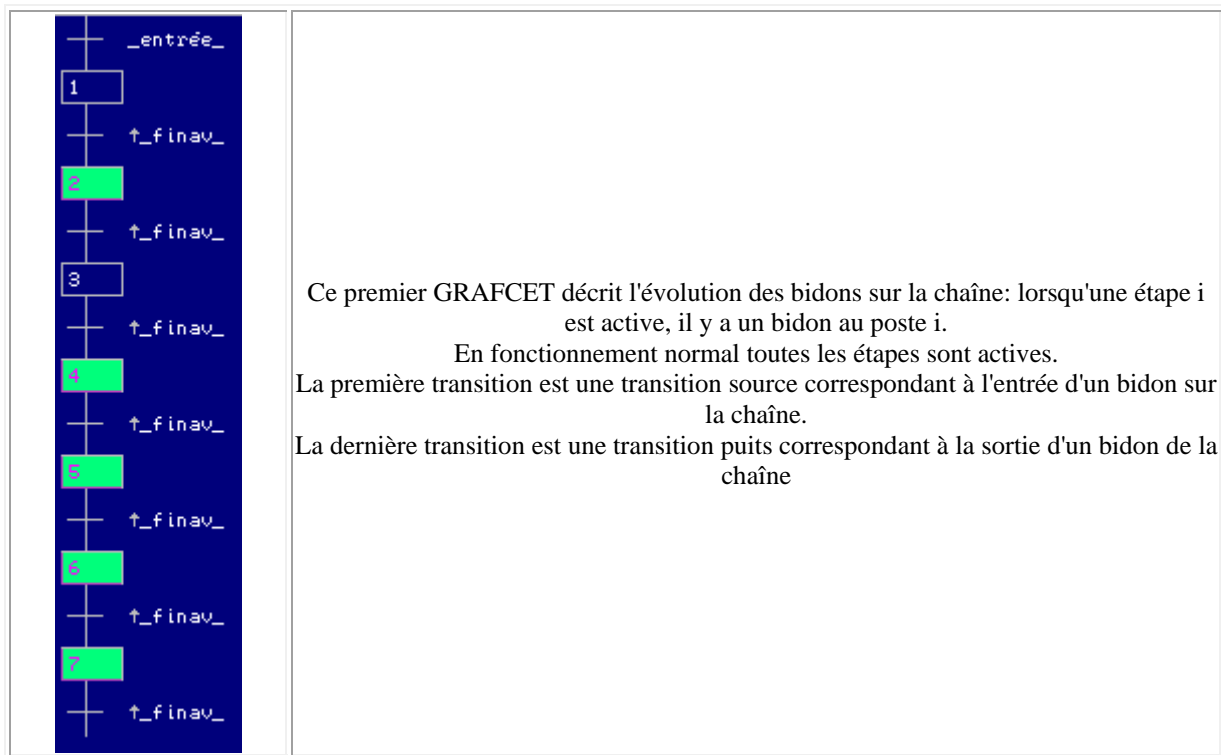


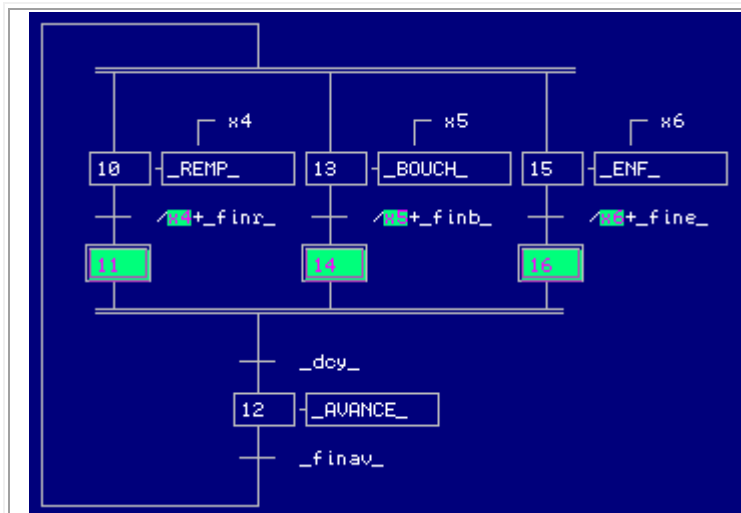
Exercice 4: Travail à la chaîne

Soit une chaîne de remplissage de bidons d'huile. Un tapis roulant se déplaçant par saccades, s'arrêtant à chaque nouvel appui de la came sur le capteur finav est alimenté manuellement sur le poste marqué entrée (de temps en temps il manque des bidons). Trois postes sont prévus : remplissage (R), bouchage (B) et enfoncement (E).



Un seul capteur ("entrée") détecte la présence d'un bidon en début de chaîne. On désire faire les 3 opérations simultanément, sauf s'il n'y a pas de bidon sous le poste. Il n'est pas nécessaire de rajouter des capteurs. On suppose que le tapis est vide lors de l'initialisation.



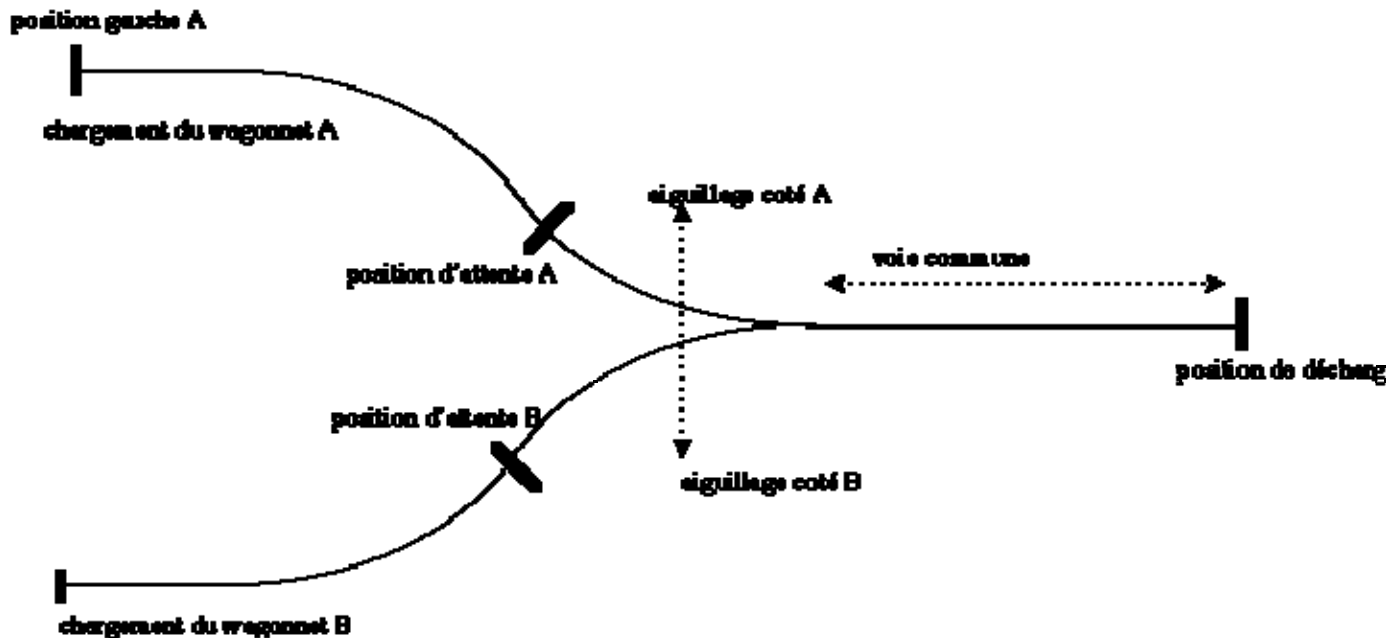


Ce deuxième GRAFCET décrit les commandes des postes 4,5,6 et la commande de l'avance de la chaîne.

Cette méthode utilise au mieux le séquençage du Grafcet, on peut maintenant rajouter des capteurs, mais qui n'auront pour fonction que de vérifier le bon fonctionnement du système. Dans tous les cas similaires, on utilisera cette démarche.

Exercice 5 : ressource (ou sémaphore)

Deux wagonnets alimentent le skip de déchargement d'un haut-fourneau en empruntant une voie commune.



Le cycle correspondant à un chariot est le suivant :

-dès que l'opérateur donne l'ordre de " départ cycle " (Dcy), le wagonnet considéré effectue automatiquement son chargement et démarre au signal fin de chargement (Fc).

-le wagonnet se dirige ensuite vers la partie commune où il s'arrête à une position d'attente si la voie commune est occupée, sinon il faut positionner l'aiguillage sur la position correcte et le chariot continue ;

-arrivé à la position de déchargement automatique, il attend le temps nécessaire avant de retourner à la position initiale (attente 10s).

-chaque déchargement de wagonnet est comptabilisé en vue d'une gestion journalière.

En utilisant les capteurs et les actions suivantes:

capteur		Commande	
départ wagonnet A	_dcya_	avance wagonnet A	_ava_
départ wagonnet B	_dcyb_	avance wagonnet B	_avb_
position d'attente A	_paa_	commande aiguillage sur pos A	_aga_
position d'attente B	_pab_	commande aiguillage sur pos B	_agb_
aiguillage en A	_paga_	retour wagonnet A	_ara_
aiguillage en B	_pagb_	retour wagonnet B	_arb_
wagonnet en C	_pdc_		
fin de déchargement	_fdch_		
wagonnet A en position de chargement	_pca_		
wagonnet B en position de chargement	_pcb_		

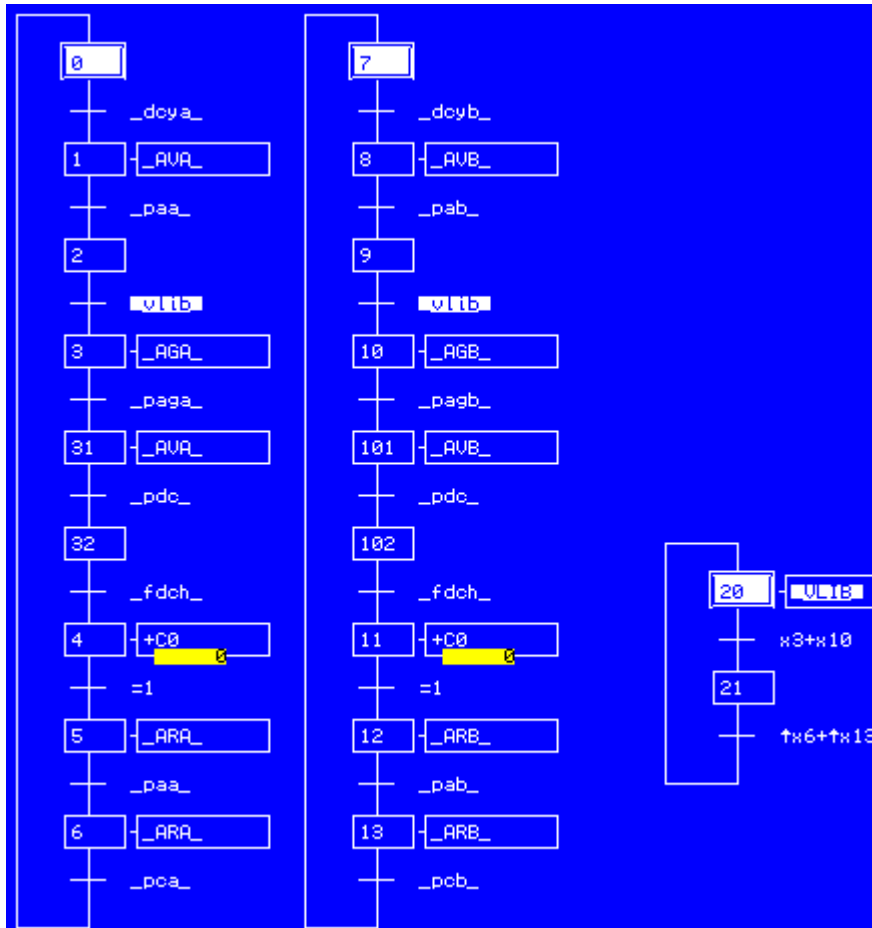
on peut donner par exemple le GRAFCET ci-dessous qui se décompose en:

-un GRAFCET pour la voie A

-un GRAFCET pour la voie B

-un GRAFCET pour déterminer si la voie est libre

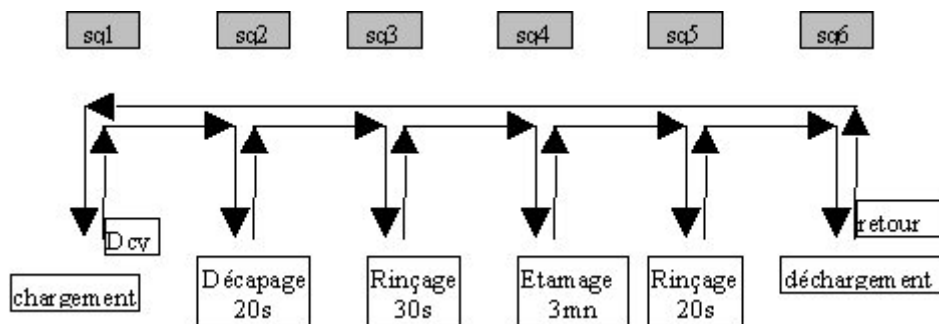
La voie commune est une ressource physique que les 2 utilisateurs ne peuvent pas utiliser en même temps. Dans les problèmes de ce type la disponibilité de la ressource commune ne peut pas être détectée par un capteur, mais uniquement par l'état du GRAFCET.



Exercice 6 : Traitement de surface

Une installation de traitement de surface comprend un chariot automoteur desservant 4 bacs, un poste de chargement et un poste de déchargement. Des capteurs `_sq1_` à `_sq6_` permettent le positionnement au dessus des différents postes (voir schéma de simulation AUTOMGEN ci-dessous)

L'opérateur après avoir accroché les pièces à traiter sur le cadre situé au point de chargement, en position basse, donne l'ordre de départ cycle. Le chariot doit alors effectuer le cycle suivant :



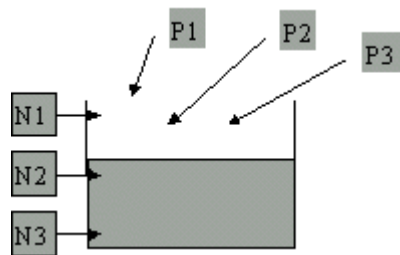
Au poste de déchargement l'opérateur décroche les pièces et renvoie le chariot avec l'ordre retour.

Donner le GRAFCET de cet automatisme.

Exercice 7: Commandes de pompes

Le niveau de liquide contenu dans un réservoir est contrôlé par trois détecteurs N1, N2, N3. L'alimentation de ce réservoir s'effectue par trois pompes P1, P2, P3 de la façon suivante :

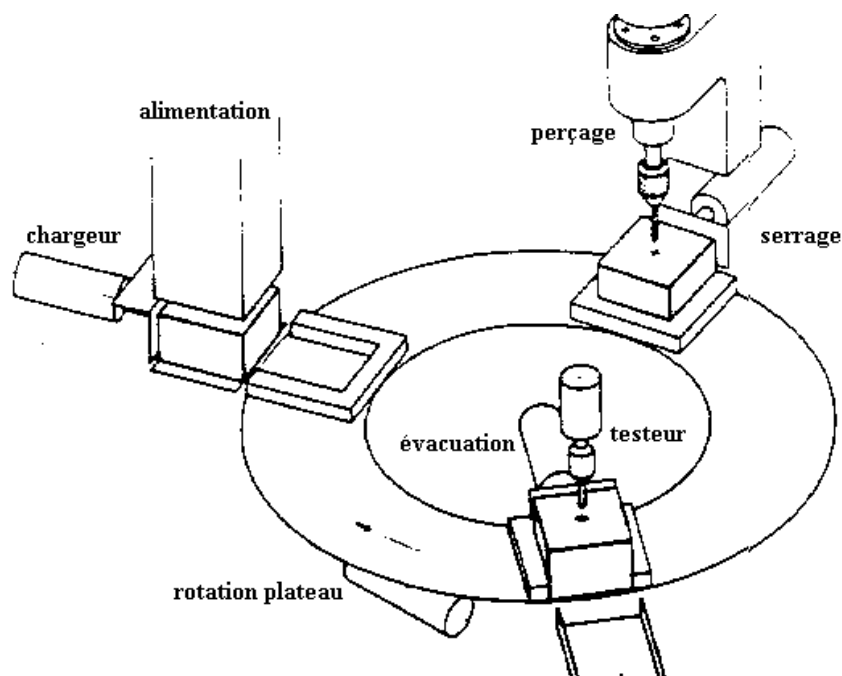
- si le niveau N1 est découvert (N1=0), une première pompe est mise en marche
- si le niveau N2 est découvert (N1=0, N2=0), une deuxième pompe est mise en marche
- si le niveau N3 est découvert (N1=0, N2=0, N3=0), la troisième pompe est mise en marche.



Le nombre de pompes en service sera égal au nombre de niveaux découverts. De plus afin d'équilibrer l'usure des pompes, celles-ci seront permutées à tour de rôle.

Donner le GRAFCET de commande des pompes.

Exercice 8: Poste de perçage



Un plateau tournant dessert 3 postes de travail :

- un premier poste de chargement
- un deuxième de perçage

-un troisième de contrôle et d'évacuation des pièces percées.

Un vérin permet la rotation de 120° du plateau supportant les pièces à usiner et son indexation, c'est à dire son blocage précis après chaque rotation.

Le contrôle du perçage s'effectue par un testeur qui doit descendre en position basse, si le trou est correctement percé. Si cela n'est pas réalisé, tout le système se bloque, testeur en position haute, de façon à ce que l'opérateur puisse enlever la pièce défectueuse avant de réarmer manuellement le système .

Donner le GRAFCET décrivant le fonctionnement de cet automatisme.

symboles utilisés pour le TD poste de perçage(fichier perceuse.sym)

capteurs	actionneurs
dcy:i0 ;départ cycle	
ch1:i1 ;chargeur avancé	_CH+_:o1 ;avance chargeur
ch0:i2 ;chargeur reculé	_CH-_:o2 ;recul chargeur
sr1:i11 ;vérin serrage avancé	_SR+_:o11 ;avance vérin serrage
sr0:i10 ;vérin serrage reculé	_SR-_:o10 ;recul vérin serrage
haut:i20 ;perceuse en haut	_MONTEE_:o20 ;montée perceuse
bas:i21 ;perceuse en bas	_DESCENTE_:o21 ;descente perceuse
te1:i7 ;testeur en bas	_TE+_:o7 ;descente testeur
te0:i8 ;testeur en haut	_TE-_:o8 ;montée testeur
ev1:i5 ;vérin évacuation avancé	_EV+_:o5 ;avance vérin évacuation
ev0:i6 ;vérin évacuation reculé	_EV-_:o6 ;recul vérin évacuation
rea:i5 ;réarmement après pièce mauvaise	
vr1:i3 ;vérin rotation plateau avancé	_VR+_:o3 ;avance vérin rotation plateau
vr0:i4 ;vérin rotation plateau reculé	_VR-_:o4 ;recul vérin rotation plateau

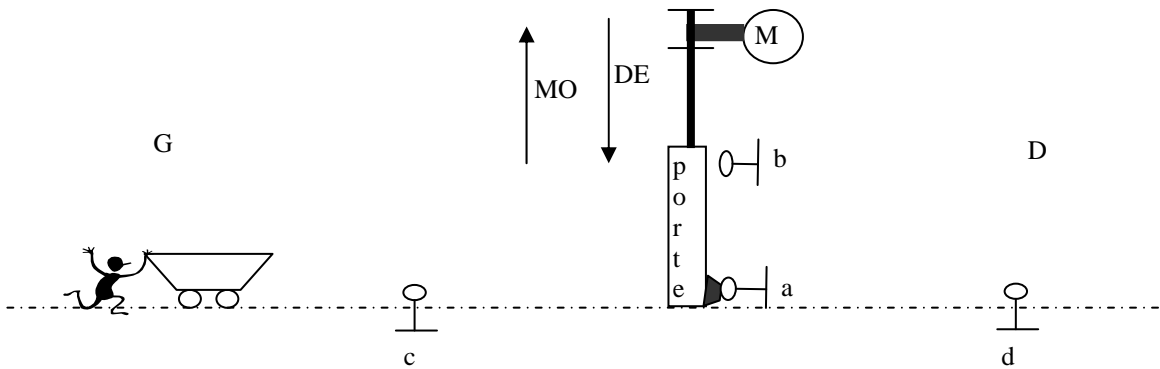
Exercice 9 : Grafcet. Commande d'ouverture de porte

Une porte soulevée par l'intermédiaire d'un groupe moto-réducteur M (à deux sens de rotation) permet le passage d'un chariot (un seul chariot existe) guidé sur rails et poussé par une personne. Le chariot doit avoir :

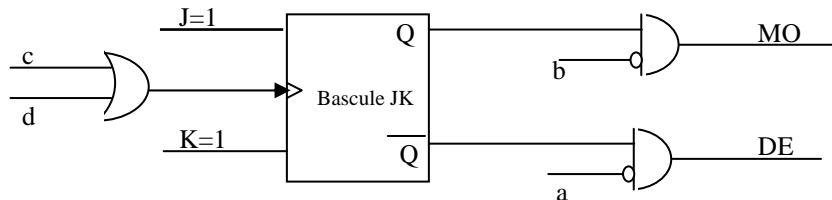
- Entrer à gauche (G) et sortir à droite (D)
- Entrer à droite (D) et sortir à gauche (G)
- Entrer à gauche (G) et sortir à gauche (G)
- Entrer à droite (D) et sortir à droite (D)

En venant de gauche (G), le chariot agit sur un contact c (impulsion) : la porte s'ouvre et le moteur effectue l'action MO. En venant de droite (D), le chariot agit sur un contact d (impulsion) : la porte s'ouvre également et le moteur s'effectue l'action MO.

En sortant le chariot agit sur c ou sur d ; la porte se ferme et le moteur effectue l'action DE. Les contacts a et b limitent les déplacements de la porte ; a=1 : porte en bas et b=1 : porte en haut



- 1- Donner le Grafset après avoir défini les différentes étapes et les différentes transitions nécessaires au cahier de charge ainsi défini.
- 2- Démontrer qu'on peut simplifier ce Grafset pour avoir 4 étapes seulement.
- 3- Démontrer qu'on peut commander l'ouverture de la porte à l'aide du schéma



électronique suivant :

A l'état initial on suppose que $Q=0$, $a=1$ et $b=0$.

Exercice 10 :

