UCL Université catholique de Louvain

MECA 2755

Automates programmables

2: Programmation du Grafcet

UCL Université catholique de Louvain

Contenu du cours

- **Introduction**
- **Les règles du grafcet**
- **La programmation du Grafcet (langage SFC)**
- **Nécessité d'une démarche**
- **Applications**

Origine

1970:

Complexité croissante des automatismes industriels

➡ Besoin de nouveaux outils de modélisation

1977:

L'AFCET(*) propose un nouvel outil d'analyse de la commande séquentielle d'un processus: le « grafcet »

GRAphe Fonctionnel de Commande des Etapes-Transitions

1982:

↳ inspiré des « réseaux de Petri (+) » : modelling systems with concurrently occurring events

Normalisation en France (NF C 03-190)

1988:

Normalisation Européenne (CEI 848) sous le dénomination de :
« Function Chart for Control Systems »

...1995:

Nouvelle norme française … intégrée dans la norme européenne relative aux langages de programmation des automates:

- Dénomination : SFC « Sequential Function Chart »
- Norme : CEI 61131-3

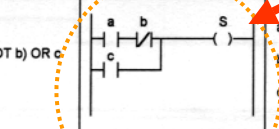
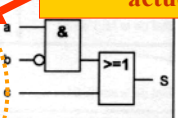
(*) Association Française pour la Cybernétique Economique et Technique

(+) Carl Adam Petri, mathématicien allemand (thèse 1962)

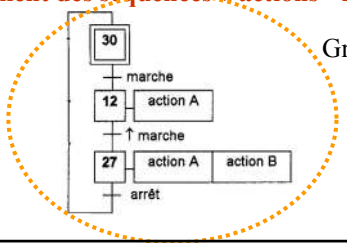
Langages de programmation

1. Encodage des actions et des réceptivités (bas niveau)

80 % des applications actuelles

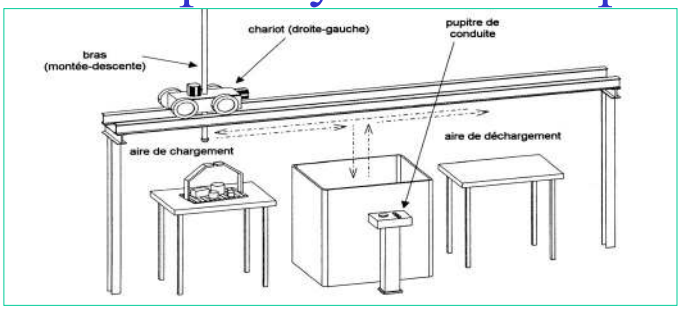
Liste d'instructions	Littéral structuré	Schéma à contacts	
(LD a ANDN b) OR c ST S ou O(U - a UN - b) O - c = -S	S := (a ANDNOT b) OR c		
langage puissant, nécessite une certaine expérience	permet les structures habituelles des langages informatiques : IF, CASE, FOR, WHILE, REPEAT	très courant, facile à mettre en oeuvre grâce à sa ressemblance avec les schémas électriques	permet de programmer rapidement des fonctions complexes par paramétrage de blocs fonctionnels
type « assembleur »	type « C »	type « électrique »	type «simulink»
Noms des langages équivalents dans la norme CEI 61131-3 (SFC, issu du Grafcet, est réservé aux fonctions séquentielles)			
IL	ST	LD	FBD

2. Enchaînement des séquences « actions - réceptivités » (haut niveau)



Grafcet (« SFC») ou langage ci-dessus

Exemple: cycle de trempe



Liste détaillée des opérations:

Liste succinte des opérations:

- L'opérateur installe le plateau et
- Le chariot se déplace jusqu'au b
- Le
- Au
- Le
- Un voyant clignote, l'opérateur
- Le chariot revient en position de

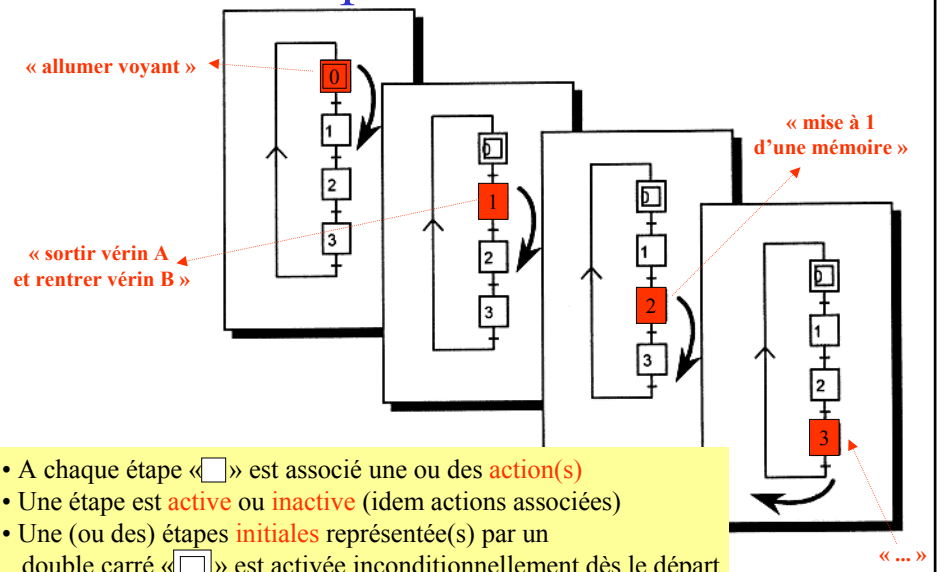
« **Actionneurs** »
« *capteurs* »

- *L'opérateur appuie sur le bouton départ.*
- **Le chariot se déplace vers la droite.**
- *Le chariot arrive au-dessus du bac.*
- **Le bras descend.**
- *Le bras arrive en position basse.*
- **Une temporisation de 8 minutes commence.**
- *Les 8 minutes sont écoulées.*
- **Le bras remonte.**
- *Le bras arrive en position haute.*
- **Le chariot se déplace vers la droite.**
- *Le chariot arrive au-dessus de l'aire de déchargement.*
- **Un voyant clignote.**
- *L'opérateur appuie sur le bouton départ.*
- **Le chariot se déplace vers la gauche.**
- *Le chariot arrive au-dessus de l'aire de chargement.*

Contenu du cours

- **Introduction**
- **Les règles du grafcet**
- **La programmation du Grafcet (langage SFC)**
- **Nécessité d'une démarche**
- **Applications**

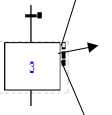
Etapes - Actions



- A chaque étape « □ » est associé une ou des action(s)
- Une étape est active ou inactive (idem actions associées)
- Une (ou des) étapes initiales représentée(s) par un double carré « ▣ » est activée inconditionnellement dès le départ

Etapes - Actions

En language PL7 (Télémécanique), on autorise la programmation de 3 types d'action pour une étape:



Actions à l'activation de l'étape

actions «impulsionnelles » exécutées une seule fois lorsque l'étape associée passe de l'état inactif à l'état actif (« sur front montant de l'étape »)

ex.: incrémentation d 'un compteur, appel d'un sous-programme, set d'une mémoire

Actions continues de l'étape

actions continues scrutées et exécutées tant que l'étape est active

ex.: opérations « <operate> » sur mots, bits, set d 'une mémoire, d'une sortie(*)

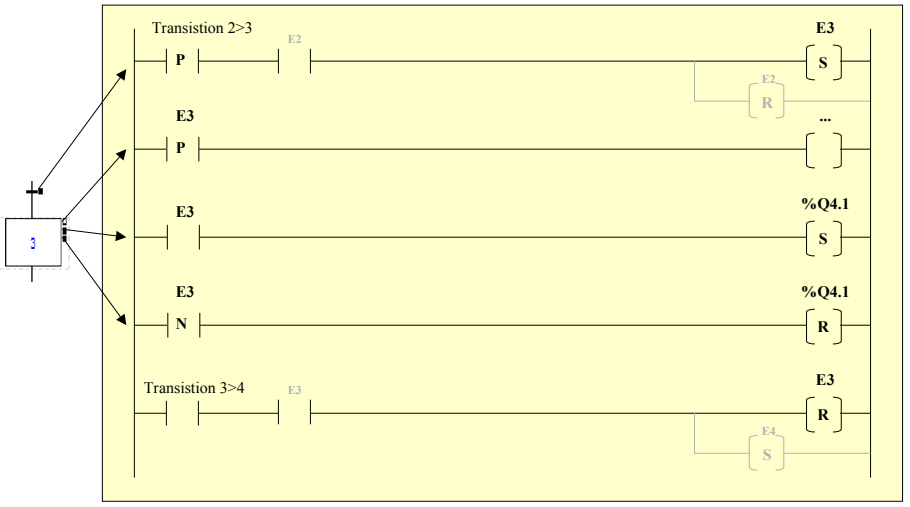
Actions à la désactivation de l'étape

actions exécutées une seule fois lorsque l'étape associée passe de l'état actif à l'état inactif («sur front descendant de l'étape»)

ex.: reset d'une mémoire, d'une sortie, d'un monostable

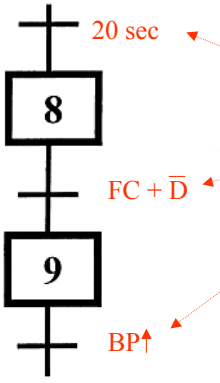
(*) pour une sortie « process » %Qi.j , on recommande de la programmer dans le « post » sur base du bit étape plutôt que dans le chart des étapes/tansitions: voir partie programmation

En ladder, ça donnerait :



Transitions - réceptivités

Les transitions relient les étapes entre-elles et doivent contenir des conditions de franchissement de la transition appelées « réceptivités »



Les réceptivités « ┼ » sont de plusieurs types:

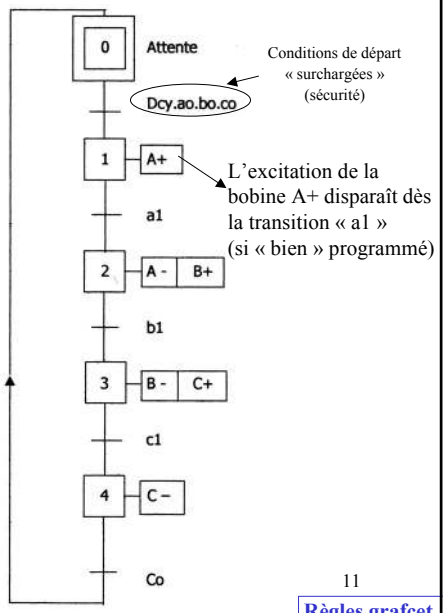
- Temps d'activité de l'étape précédente
- Condition logique (booléenne)
- Apparition Front montant ou descendant
- ...
- Toujours vraie (« =1 »)

Etapes - Transition

A+ {A-,B+}, {B-, C+}, C-

Toujours respecter l'alternance
étapes-transition,
que le grafcet soit linéaire,
à séquences multiples,
avec sous-programmes,
...

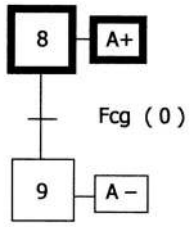
A chaque cycle automate,
le programme (en interne) met à jour:
- une table d'activité des étapes:
  - étapes atives
  - étapes à activer
  - étapes à désactiver
- une table de validité des transitions:
  - transitions en aval des étapes de la table d'activité



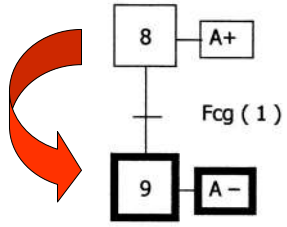
Franchissement d'une transition

Il y a deux conditions impératives pour franchir une transition:

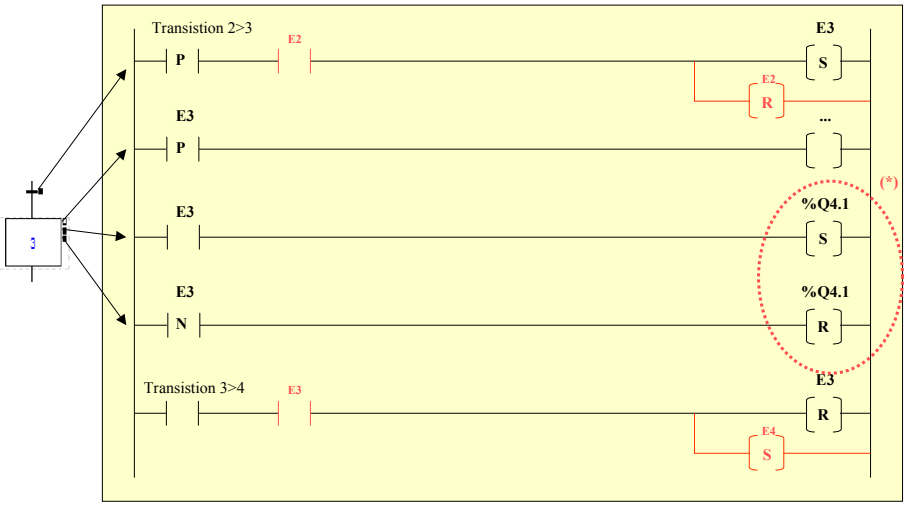
1. La réceptivité associée
   doit être vraie (… évidemment !)
2. L'étape en amont doit être active
   (… moins évident en pratique)



Les action de l'étape amont ne sont plus scrutées après le franchissement de l'étape
mais les sorties affectées restent mémorisées !!!!
Exemple :
la mise à 1 d 'une sortie « M1 »
à l'étape 8, le reste à l'étape 9.
Elle doit donc être remise à zéro
(idéalement à la désactivation de l'étape 8)



En ladder, çà donnerait :

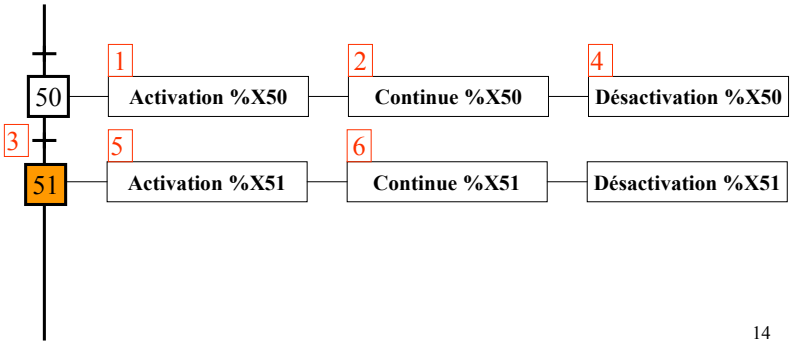


(*) pour une sortie « process » %Qi.j , on recommande de la programmer dans le « post » sur base du bit étape plutôt que dans le chart des étapes/tansition: voir partie programmation

Ordre d'exécution des actions

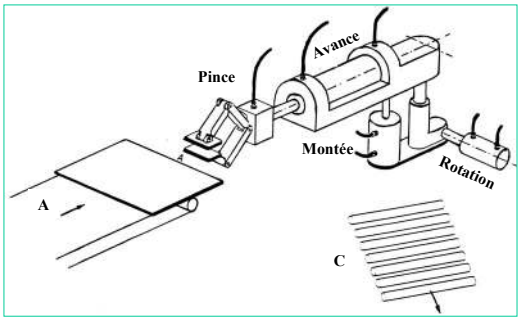
Lorsque l'étape **51** est activée (suite à la validation de la transition), l'ordre d'exécution des actions est le suivant:

1. Actions à la désactivation de l'étape 50
2. Actions à l'activation de l'étape 51
3. Actions continues de l'étape 51



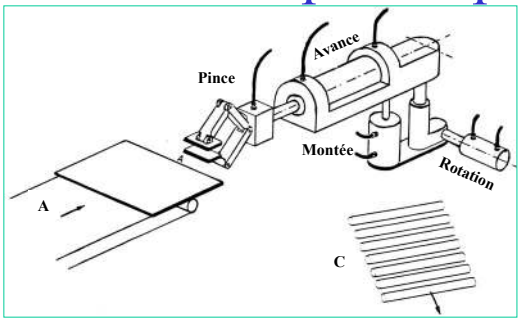
Un exemple simple : grafcet linéaire

Sur Départ cycle, transférer une plaque du convoyeur A au convoyeur C

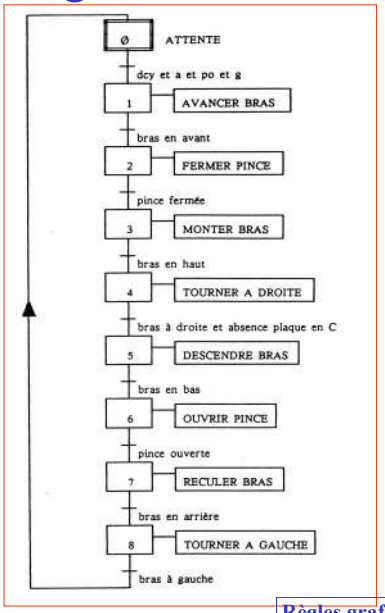


Actions		Informations	
Avance du bras	AV	Départ cycle	dcy
Recul du bras	AR	Bras en avant	av
Fermeture pince	FP	Bras en arrière	ar
Ouverture pince	OP	Pince fermée	pf
Descente bras	DB	Pince ouverte	po
Montée bras	MB	Bras en bas	b
Rotation droite	RD	Bras en haut	h
Rotation gauche	RG	Bras à droite	d
		Bras à gauche	g
		Présence plaque en A	a
		Présence plaque en C	c

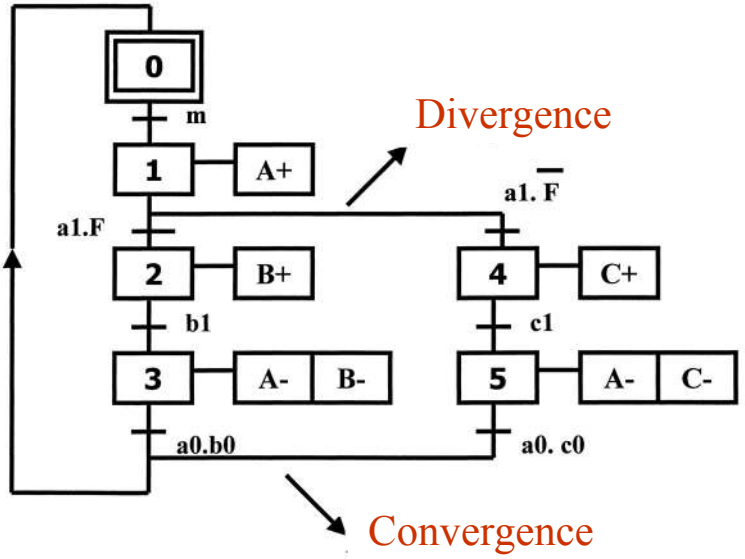
Un exemple simple : grafcet linéaire



Départ cycle	dcy
Bras en avant	av
Bras en arrière	ar
Pince fermée	pf
Pince ouverte	po
Bras en bas	b
Bras en haut	h
Bras à droite	d
Bras à gauche	g
Présence plaque en A	a
Présence plaque en C	c

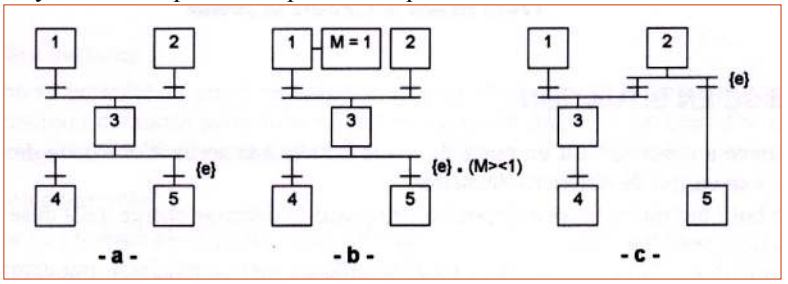


Structure en « OU »



Structure en « OU »

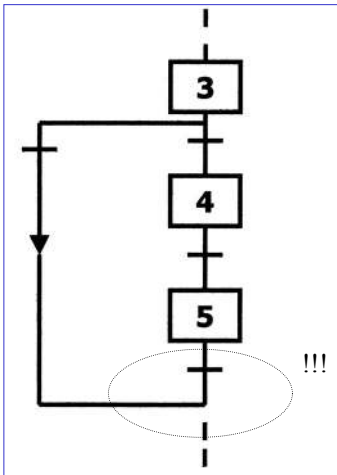
Tout comme en programmation classique, éviter des ossatures « boîteuses » en termes d'organisaiton des étapes et des équations logiques
Il y a des exemples « à la pelle » tel que celui-ci :



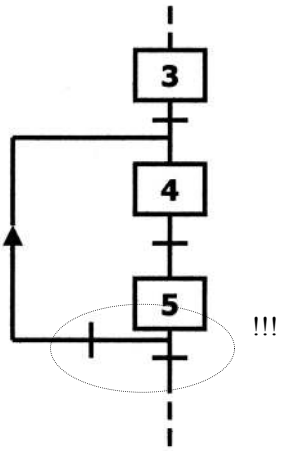
Si le grafcet ne doit pas évoluer de X1 à X5 même si {e} est vraie:
- la solution (b) fonctionne mais est « bricolée »
- la solution (c) est propre :
  - Les séquences obligées sont explicitement traçées (et elles-seules)
  - La réceptivité en amont de l'étape X5 se limite à l'expression réellement utile.

Cas particuliers du « OU »

Saut d'étapes



Reprise de séquences

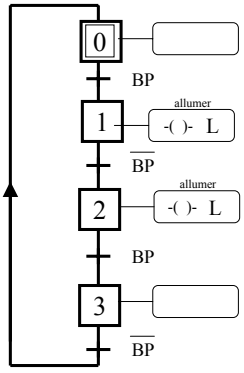


Exercice: le télérupteur

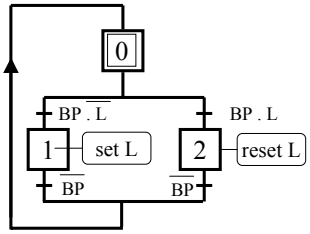
Enoncé:

Une impulsion sur un bouton poussoir BP enclenche l'éclairage « L » d'un hall. Une deuxième impulsion sur le même bouton éteint L, … et ainsi de suite.

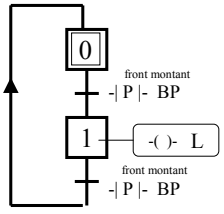
Solution 1: « lourde »



Solution 2: « réfléchie »
(OU et « set/reset »)

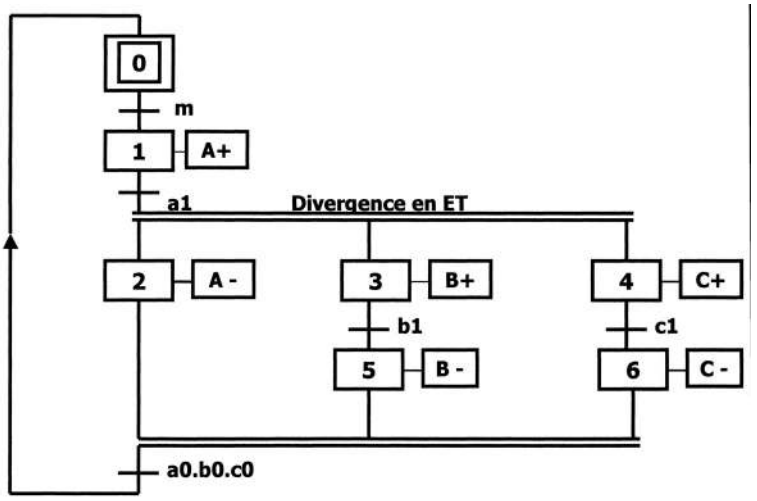


Solution 3: « astucieuse »
(Front montant BP)



NB: ceci est un exercice;
on n'utilise (bien sûr) pas
un automate pour ce genre d'opération !

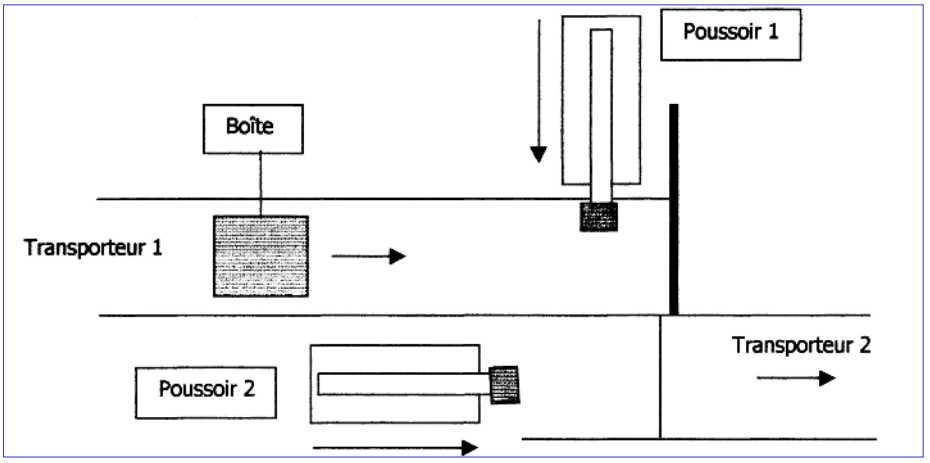
Structure en « ET »



- **Les branches du « ET » évoluent à leur rythme**
- **Les étapes 2, 5 et 6 doivent être actives pour que la transition a0.b0.c0 puisse être franchie.**

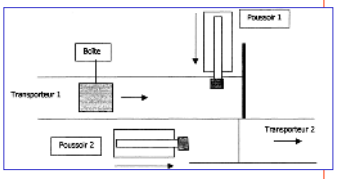
Un petit exemple (« ET »)

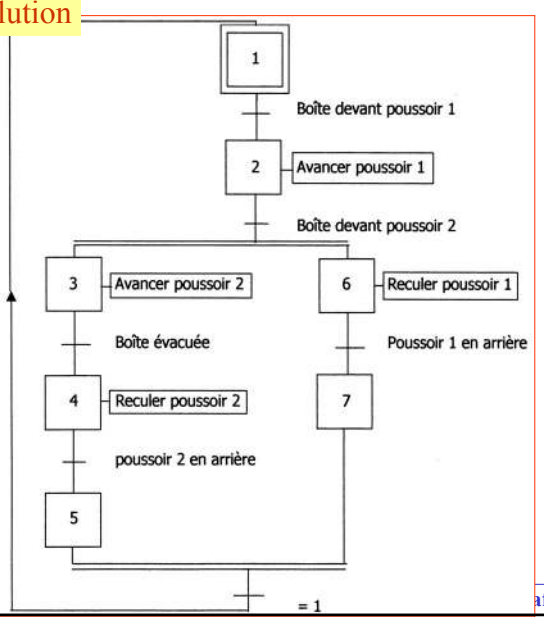
Enoncé: Transfert de boîtes du transp. 1 vers transp. 2



Un petit exemple (« ET »)

Transfert de boîtes: une solution

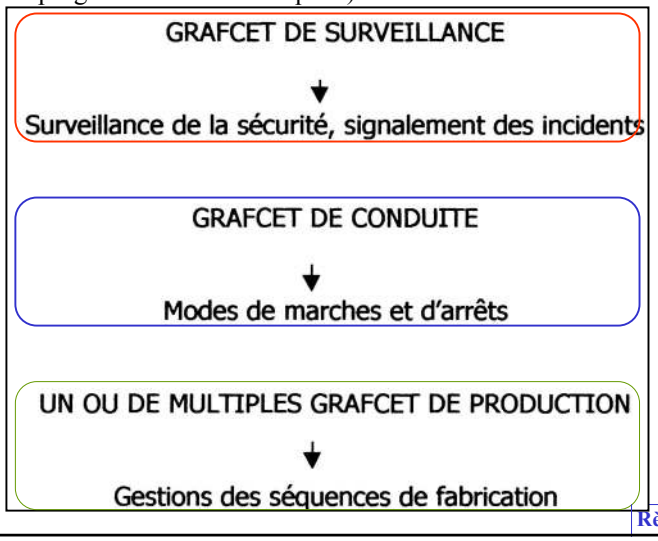




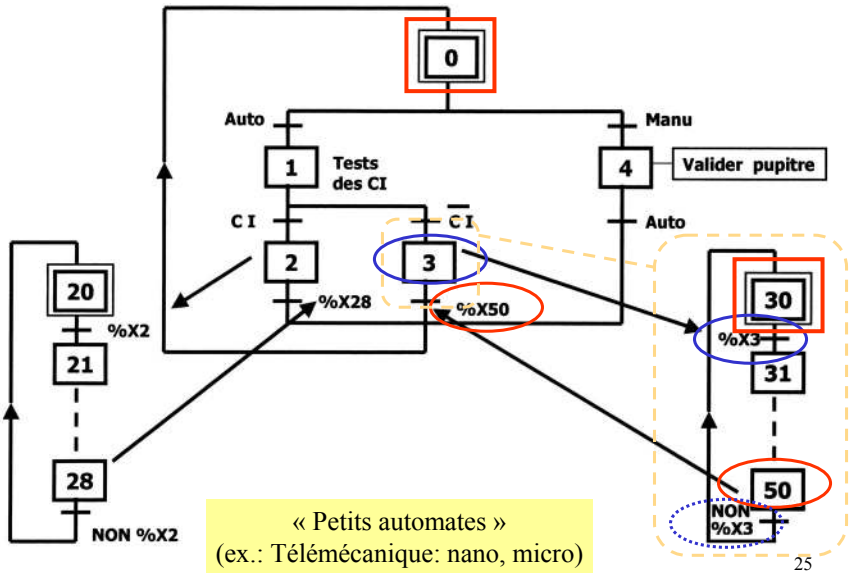
Synchronisation et macro-étapes

Pour un processus assez complexe, il faut d'abord une phase d'analyse pour décomposer le processus en termes de tâches et de hiérarchie (comme en programmation « classique »).

Typiquement :



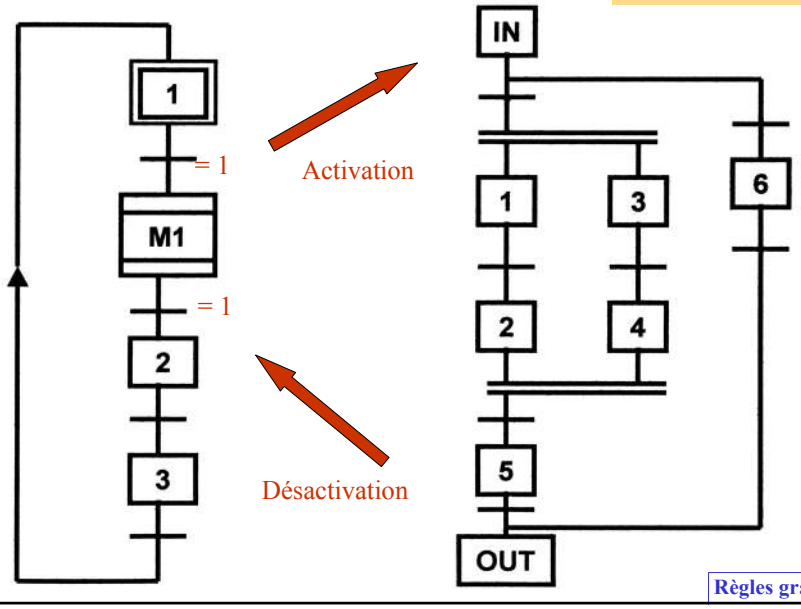
Synchronisation : principe



« Petits automates »
(ex.: Télémécanique: nano, micro)

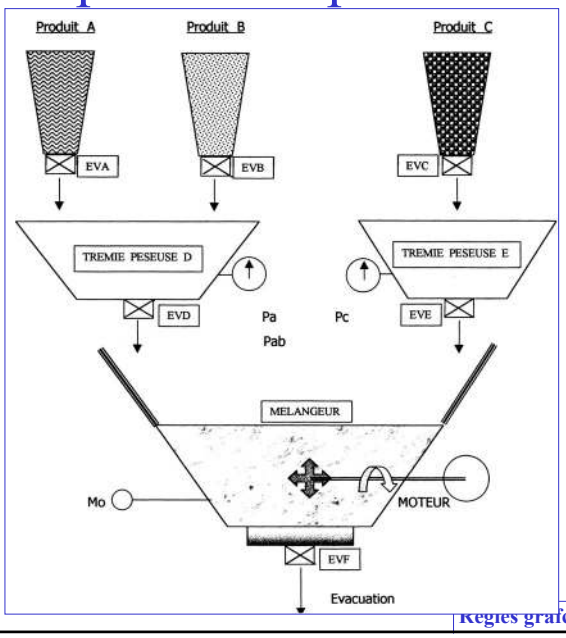
Macro-étapes (Tsx)

symbole: 

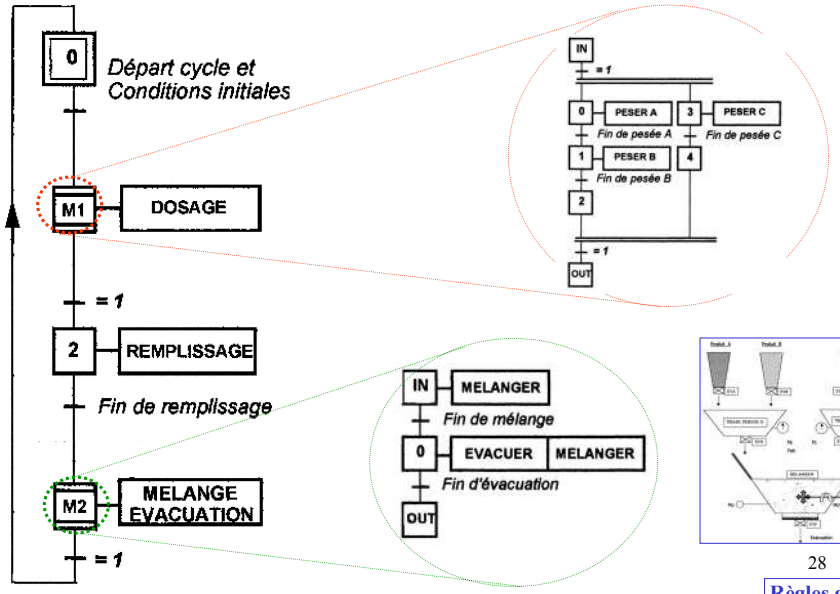


Macro-étapes : exemple

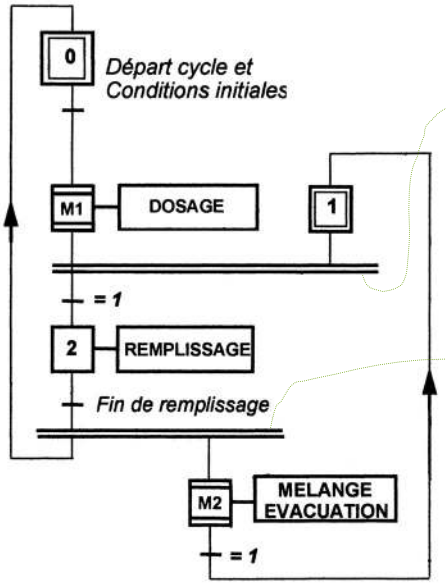
Soit un process qui doit:
1. doser 3 produits A, B, C
   (A et B dosés ensemble :
   A puis A+B,
   C, dosé séparément)
2. mélanger les trois produits
3. évacuer les produits



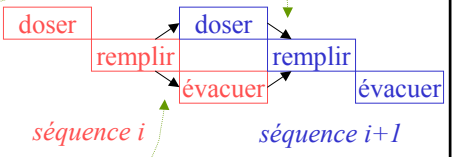
Macro-étapes : exemple

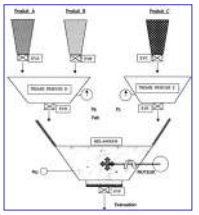


Macro-étapes : exemple

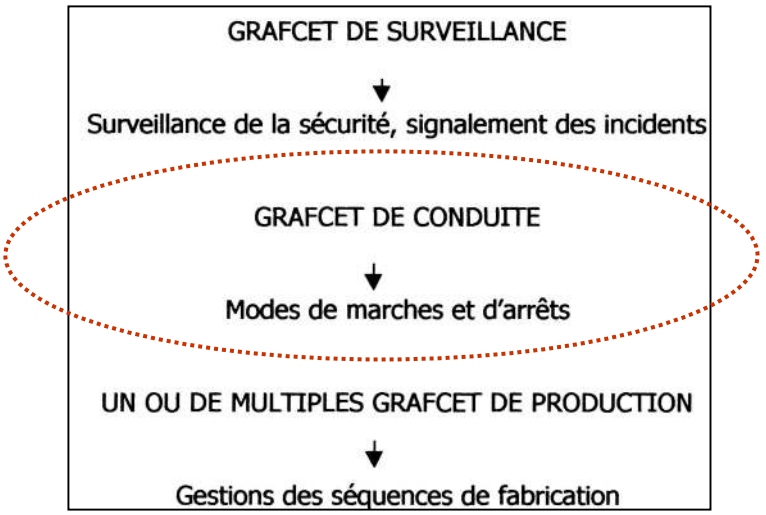


Solution imbriquée plus efficace





Graphe de conduite « Gmma »

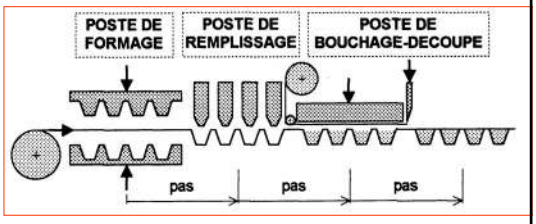


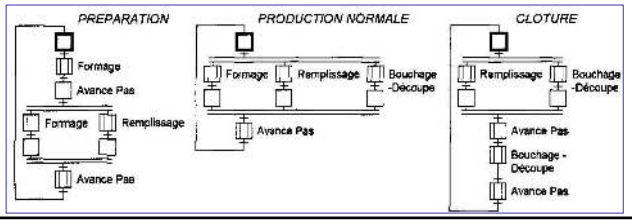
Graphe de conduite « Gmma »

Principe:

Pour gérer les modes d'arrêt, de démarrage (à froid, à chaud), ou de façon plus générale, pour organiser les différents modes de marche de l'automate (manuel, automatisé, progressif), on a recours à un grafcet de Gestion des Modes de marche et d 'arrêts (ou « GMMA »).

Exemple: moulage et remplissage de pots de yaourt





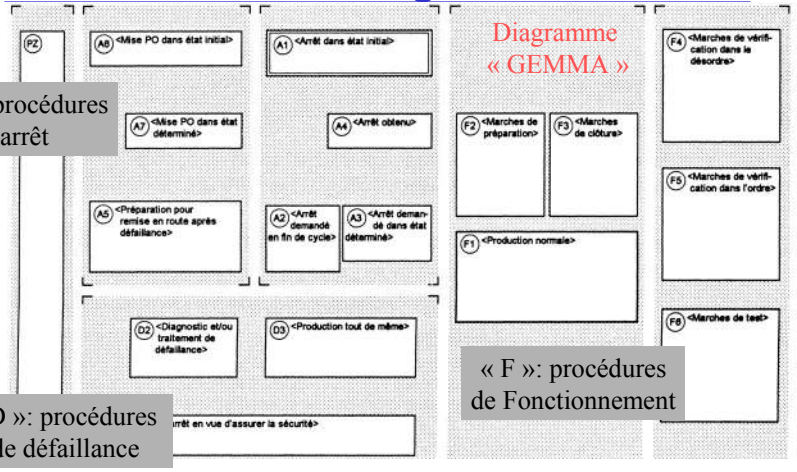
Grafcet Gmma / diagramme Gemma

« A »: procédures d'arrêt

« D »: procédures de défaillance

Diagramme « GEMMA »

« F »: procédures de Fonctionnement



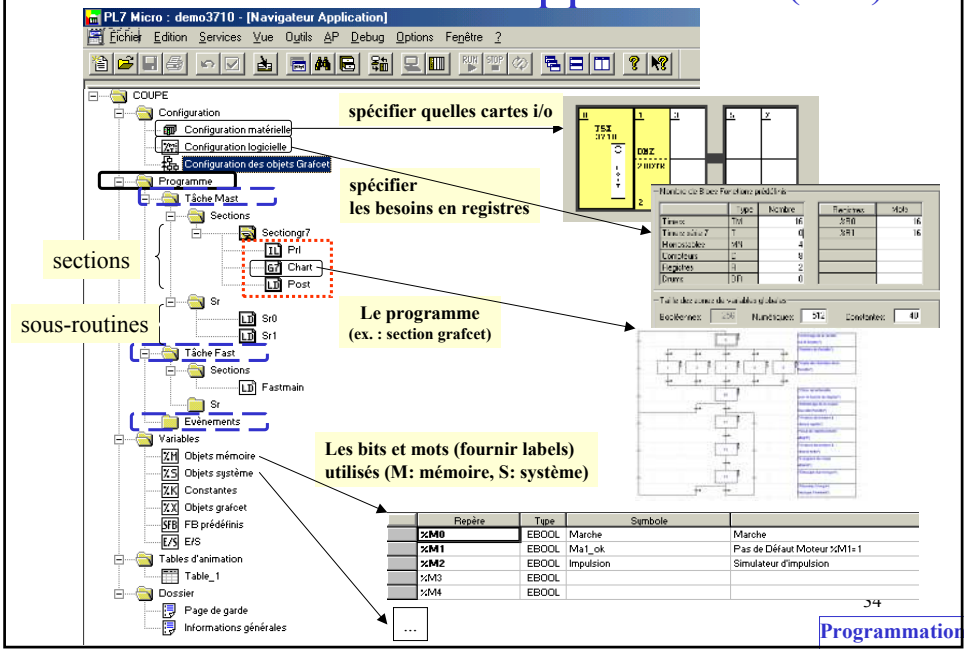
… a donné lieu au GEMMA « Guide d 'Etude des Modes de Marche et d'Arrêts »:
il se présente sous forme d'un diagramme qui contient tous les modes rencontrés en pratique:

- march de préparation, marche de clôture, production normale, arrêt dans état initial, défaillance, ...

Contenu du cours

- **Introduction**
- **Les règles du grafcet**
- **La programmation du Grafcet (langage SFC)**
- **Nécessité d'une démarche**
- **Applications**

Création d'une application (PL7)



La tâche grafcet: les 3 traitements

Prl: traitements préliminaires

- Gestion des reprises secteur
  (ex.: dém. à froid, à chaud)
- Prépositionnement du graphe lors d'incidents,
  de procédure d 'entretien, ...
  (forcer l'activation de telle ou telle étape:
  mise à 0 ou 1 de %Xi)
- Gel d'étapes = geler les étapes actives,
  quoi qu 'il arrive sur les réceptivités en aval

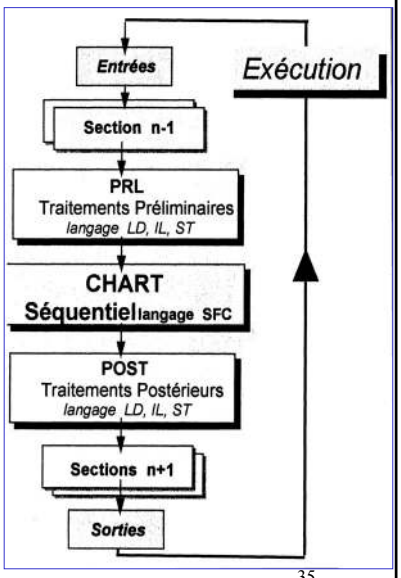
Chart

- Ossature séquentielle de l'application
  (en Grafcet, Ladder, Listes d 'instructions, …)

Post: traitements postérieurs

- Programmation d'événements indépendants
  du process séquentiel
- Programmation de la logique de sortie :
  « *On recommande de programmer les actions agissant directement sur le process dans le Post* »
  Exemple (en ladder) : programmation de l'action continue de l'étape grafcet n° 3

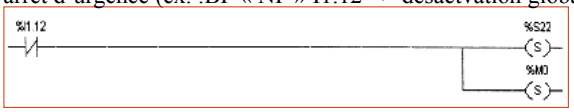
|---|%X3|----------(%Q4.2)------|



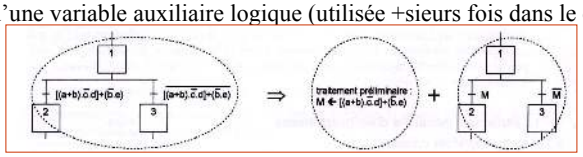
Pré et post : exemple

Prl: traitements préliminaires

- Traitement arret d'urgence (ex. :BP « NF » I1.12 => désactvation globale du chart)

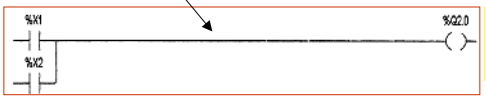


- Précalcul d'une variable auxiliaire logique (utilisée +sieurs fois dans le chart)



Post: traitements postérieurs

- Positionnement des sorties – raison : une sortie sera de préférence positionnée par une seule équation (plus prudent si plusieurs grafcet au sein du même programme)



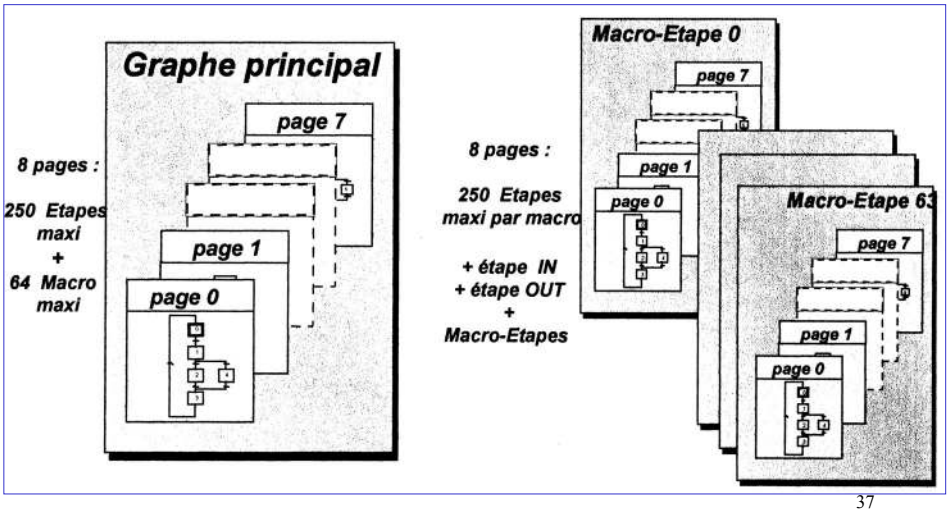
… sécurité : car une sortie « agit » sur l'environnement

Traitement préliminaire
Traitement du grafcet g-1
Traitement du grafcet g
Traitement du grafcet g+1
Traitement des sorties
dans un ordre quelconque

Ordres de grandeur

Automate « Premium »

(nano - micro - premium - quantum)



Ordres de grandeur

	Micro 37-10		Micro 37-20		Premium	
	défaut	max	défaut	max	défaut	max
Etapes du Graph principal	96	96	128	128	128	250
Macro-étapes	0	0	0	0	8	64
Etapes de macro-étapes	0	0	0	0	64	250
Total d'étapes	96	96	128	128	640	1024
Etapes actives simultanément	16	96	20	128	40	250
Transitions valides simultanément	20	192	24	256	48	500

Pas de macroétapes disponibles sur les TSX micros: utiliser la synchronisation

Objets spécifiques du langage SFC

Bits:

- Les bits « étapes » : %Xi (i = n° étape) 250 max.
- Les bits « macro-étapes » : %XMj (j = n° macro) 64 max.
- Les bits « étapes de macro » : %Xj.i (étape i de la macro j) 250 max.
- Les bits « étapes d'entrée macro » : %Xj.IN (entrée de la macro j).
- Les bits « étapes de sortie macro» : %Xj.OUT(sortie de la macro j).

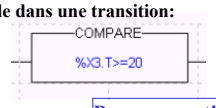
Ces bits:

- passent à 1 automatiquement en cas d'activité
- sont indexables (ex.: %X4[%MW3])
- peuvent être testés dans toutes les tâches
- peuvent être écrits dans le PRLmast uniquement (prépositionnement du grafcet)

Mots (temps d'étape)

%Xi.T : temps en 1/10e de seconde de l'étape i
%Xj.i.T : temps en 1/10e de seconde de l'étape i de la macro j

…utile dans une transition:



Bits système dédié au SFC (pour info)

%S21 : initialisation

%S21 = 1 : désactivation de toutes les étapes actives
puis activation des étapes initiales

=> mis à 1 par programme dans le Prl uniquement
=> mis automatiquement à 1 par le démarrage à froid (%S0 = 1)
=> mis à zéro par le système en fin de préliminaire (comme %S0).

%S23 : gel/dégel

%S23 = 1 : Les grafcets sont figés quelle
que soit la valeur des réceptivités: « on ne bouge plus »

=> mis à 1 par programme dans le Prl uniquement
=> mis à zéro par le système en fin de préliminaire (comme %S0).

%S24 : désactivation de macro étapes

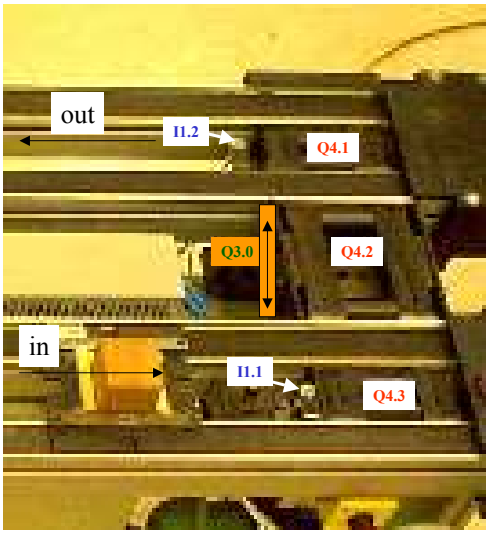
%S24 = 1 : Les macro étapes (4 maximum) renseignées
dans les « mots système %SW22 à %SW25 »

=> mis à 1 par programme dans le Prl uniquement
=> mis à zéro par le système en fin de préliminaire (comme %S0).

%...

Exercice d'assouplissement

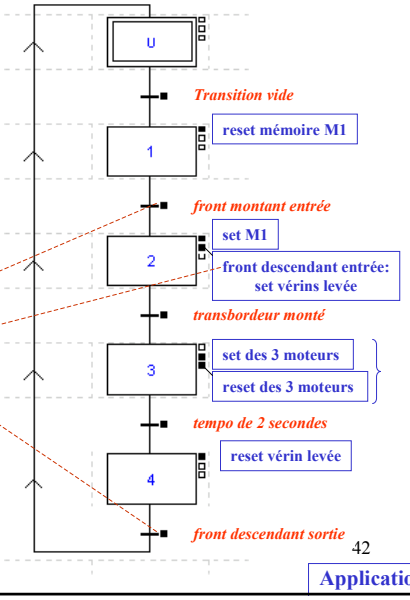
Transfert palet (labo)



Exercice d'assouplissement

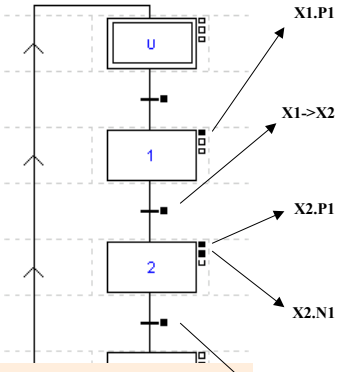
Grafcet

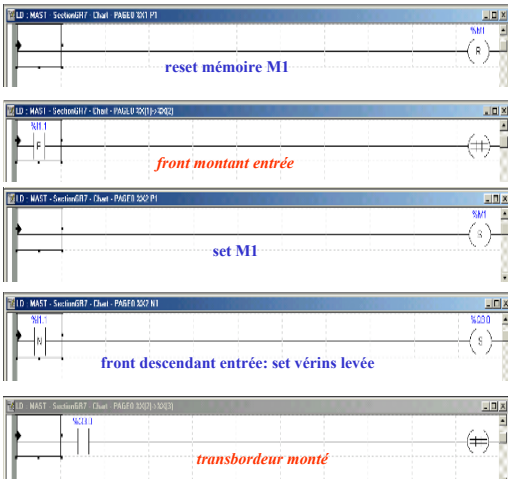




Exercice d'assouplissement

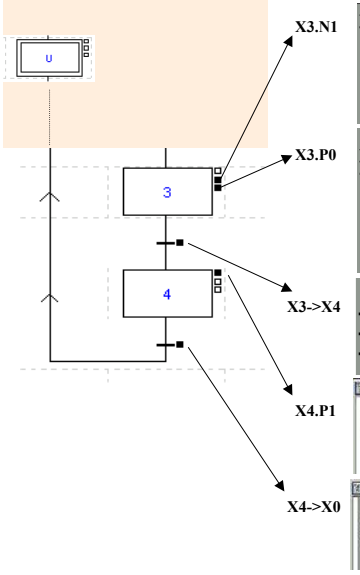
Actions et Transitions « chart »

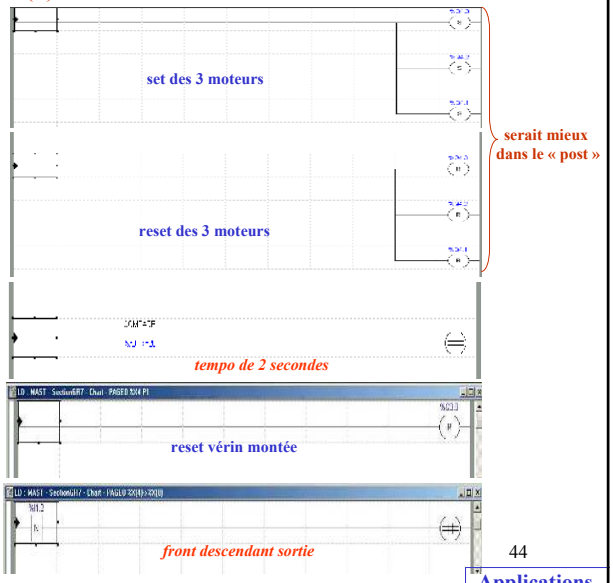




Exercice d'assouplissement

Actions et Transitions « chart » (2)





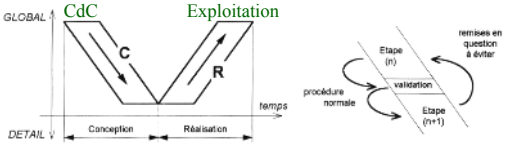
Contenu du cours

- **Introduction**
- **Les règles du grafcet**
- **La programmation du Grafcet (langage SFC)**
- **Nécessité d'une démarche**
- **Applications**

Cycle de vie d'un « système »

Système $\triangleq$ machine spécifique nécessitant un automatisme, suite à un besoin « utilisateur »

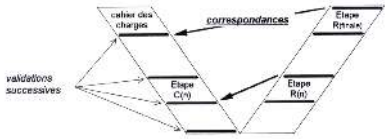
Projet de conception et réalisation d'un « système »



Abscisses = temps
Ordonnées = finesse de l'étude

C : du « global » (fonctions) aux choix technologiques retenus (analyse descendante)
R : des éléments unitaires … à la mise en exploitation (analyse ascendante)
+ étapes de validations

Cohérence du projet



Le travail réalisé et validé à l'étape R(n) doit correspondre aux décisions prises à l'issue de l'étape C(n).
=> le système final sera sera conforme au Cahier des Charges (= étape « C(0) »)

La réalisation de l'automatisme via le Grafcet (>< SFC) suivra – idéalement - ce même cheminement en « V », avec des formes et des syntaxes adaptées aux interlocuteurs

Construction des grafcets

Avec ou sans méthode

1. Sans méthode = « A l'intuition »

Envisageable pour des automatismes simples, à faible taux de parallélisme entre tâches.
Limitations et problèmes :

- programmes peu lisibles, réceptivités pas toujours cohérentes …
- re-programmation compliquée
- souvent le résultat d'essais/erreurs qui s'en ressentent dans la version finale.

2. Avec méthode = approche basée sur les fonctions

Le CdC énonce le besoin en termes de finalité : il exprime les fonctions à assurer
(ex. trier des tubes, assembler des interrupteurs, poinçonner et cisailler des flans, …)
auxquelles sont associées des contraintes (ex.: sécurité des personnes, du matériel, de l'environnement, interaction avec les opérateurs, fonctions de communications, …)

Parallélisme entre

- l'analyse de la structure du système => domaine de la conception mécanique
- l'analyse des tâches à accomplir => domaine des automatismes séquentiels

Construction des grafcets

(suite) : Avec méthode = approche basée sur une analyse fonctionnelle

=> Conception mécanique



- La décomposition fonctionnelle conduit à choisir des constituants physiques
  (un vérin, un capteur inductif, une butée, un convoyeur …)
- Les constituants sont « localisés » dans l'espace (=> layout d'installation)

=> Automatisation



- La décomposition fonctionnelle ou « en tâches » définissent des actions
  (actionner un vérin, mise en route d'un moteur, déclenchement d'une alarme …)
- Les tâches et les actions sont « synchronisées » dans le temps (=> grafcet)

Construction des grafcets

Relation entre «Fonction / Tâches / Actions

=> A la notion de fonction, sous-fonctions, on associe resp. (en grafcet) les notions de coordination de tâches et de tâches

=> A chaque tâche (resp. coordination de tâches), on associera

- un (ou plusieurs) « grafcet de tâche » qui exécute les tâches associées
- resp. un « grafcet de coordination de tâches » qui organise la chronologie des tâches

=> Une tâche consiste en une action ou une succession indissociable d'actions

=> A chaque action correspond une ou plusieurs étapes (indissociables) d'un grafcet.

Exemples

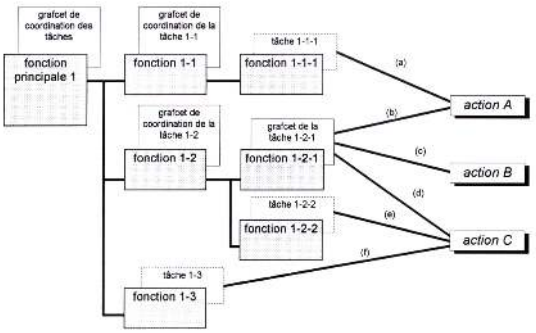
- Fonction ⇔Coordination de tâches
  - ⇒Trier des pièces
- Sous-fonctions ⇔ tâches
  - ⇒ réceptionner les pièces
  - ⇒ transporter les pièces
  - ⇒ sélectionner l'emplacement
- Actions
  - ⇒ actionner un vérin
  - ⇒ actionner un convoyeur
  - ⇒ soulever un clapet

Technologie-indépendant … analyse fonctionnelle

Technologie-dépendant … avant-projet

Construction des grafcets

Relation entre «Fonction / Tâches / Actions (suite)



(a) : action A directement associée à une étape du grafcet de coordination de la tâche 1-1

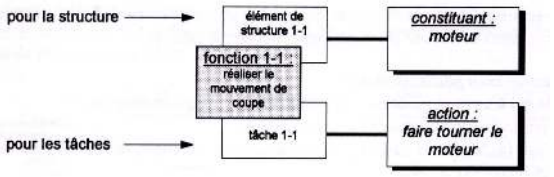
(b) : action A associée à une étape du grafcet de la tâche 1-2-1

(b), (c), (d) : la tâche 1-2-1 nécessite 3 actions différentes « indissociables », coordonnées par grafcet

(d), (e), (f) : C participe à la réalisation de trois tâches.

Construction des grafcets

Indépendance entre structure et tâche



Mais, on peut aussi avoir :

- Plusieurs constituants pour une tâche élémentaire:

  ex.: évacuer une pièce = ouvrir une porte (poussoir 1) et chasser la pièce (poussoir 2)

- Un constituant pour plusieurs tâches:

  ex.: un vérin A peut servir (A+) à « alimenter » un poste de coupe et (A-) à évacuer la pièce

- Une action pour plusieurs tâches:

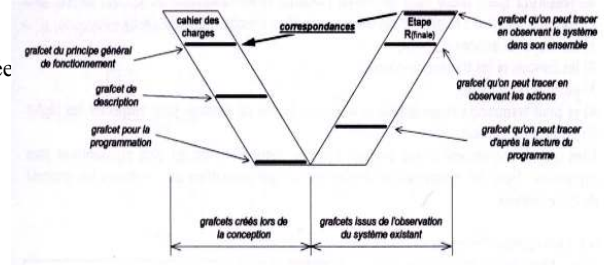
  voir figure : l'action « lever le clapet » servira tantôt à la tâche « transférer à la goulotte supérieure (a), tantôt à la tâche « laisser passer dans la goulotte courante (b) »



Construction des grafcets

Grafcet et cycle de vie

Chaque étape de la conception (« C ») sera (idéalement) accompagnée d'un grafcet dont la forme et la syntaxe seront adaptées aux interlocuteurs !



On distingue :

1. Le grafcet de fonctionnement : il se base sur les fonctions et peut être écrit (en tous cas compris) par tout interlocuteur avisé (ingénieur, technicien, directeur ?)

   => Il ne doit faire intervenir aucun élément technologique



2. Le grafcet de description : plus détaillé que le précédent, il peut est rédigé « à l'intuition » par rapport aux fonctions à remplir et au séquencement

   => Il fait intervenir les éléments technologiques choisis lors de la phase de conception

3. Le grafcet de programmation : c'est le Cahier des charges du programmeur, lequel affinera encore et fera les choix « algorithmiques » de bas niveau

   => Il est le fruit de l'analyse fonctionnelle détaillée et de ce fait distingue les grafcet de coordination de tâches et les grafcet de tâches.

Contenu du cours

- **Introduction**
- **Les règles du grafcet**
- **La programmation du Grafcet (langage SFC)**
- **Nécessité d'une démarche**
- **Applications**

Problème 1: poste de tri

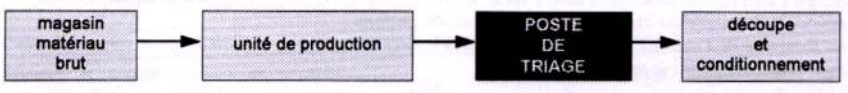
Mise en situation

Enoncé : Concevoir un poste de triage installé à la sortie d'une unité de production de tubes en carton de différents diamètres

But : mettre les tubes à disposition du poste de découpe en aval

Principe : la pesée d'un tube à l'entrée du poste de triage permet d'identifier son type : un type par diamètre.

Il existe 7 types de tubes et se présentent dans un ordre aléatoire.



Fonction principale

Transférer les tubes de type « i » vers l'emplacement « i »

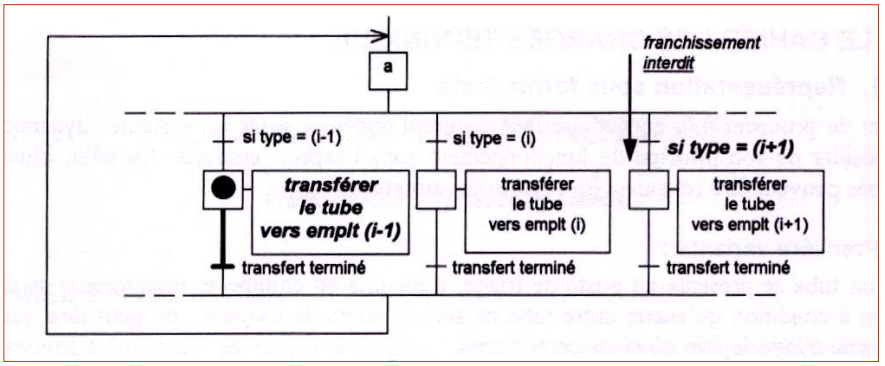


Problème 1: poste de tri

Cahier des charges techniques : +sieurs variantes

1. Triage successif : un tube est pris en charge si aucun tube n'est en cours de triage

En « grafcet », cela donnerait :

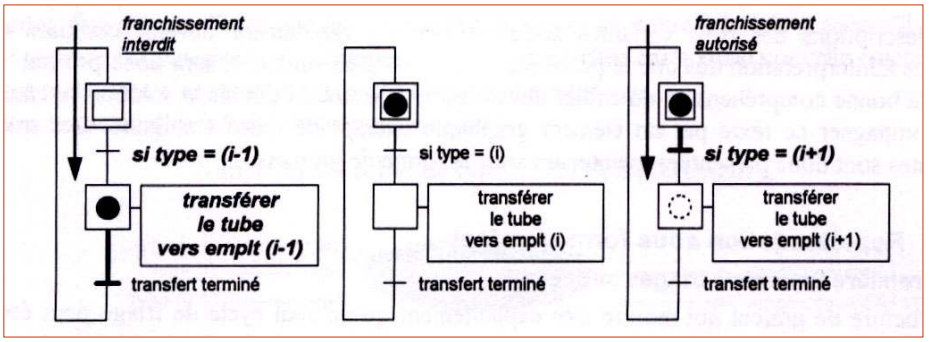


Problème 1: poste de tri

Cahier des charges techniques : +sieurs variantes

2. Triage simultané avec limite imposée : un tube qui se présente est pris en charge sauf si le tube en cours est du même type

En « grafcet », cela donnerait :

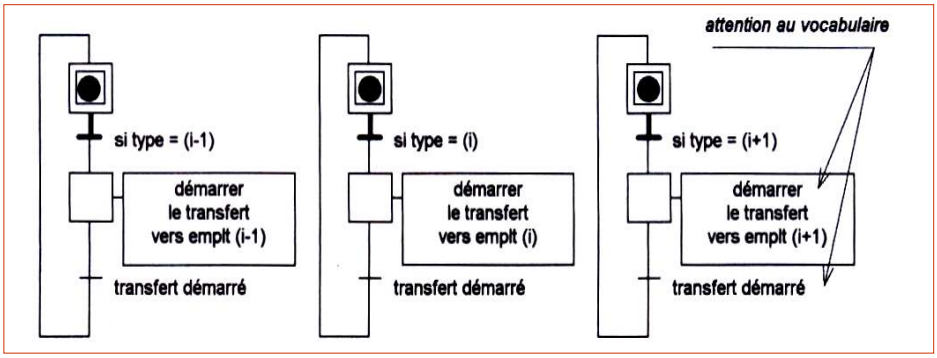


Problème 1: poste de tri

Cahier des charges techniques : +sieurs variantes

3. Triage simultané : triage simultané de 2 tubes autorisé

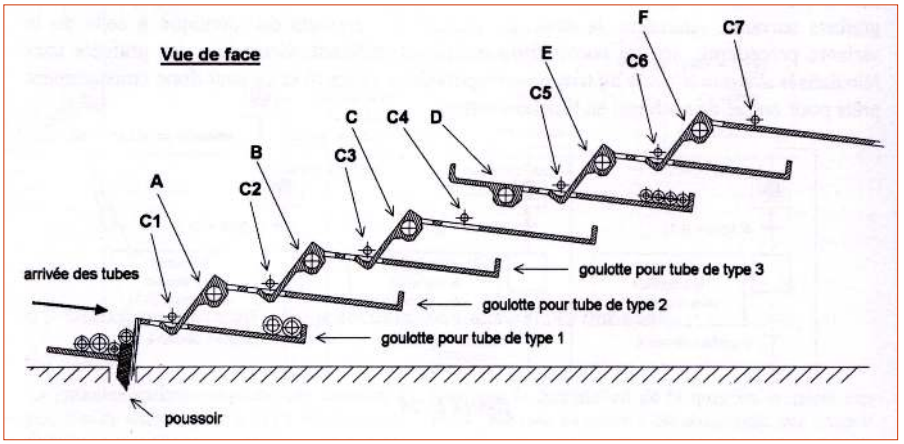
En « grafcet », cela donnerait :



Validation d'une solution retenue (avant-projet) : version 1 – triage non simultané.

Problème 1: poste de tri

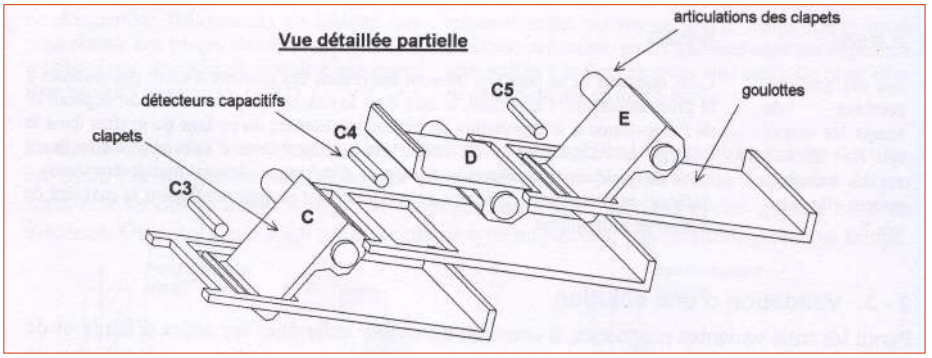
Schéma d'ensemble – avant-projet mécanique



… Un tube 4 est en cours de triage

Problème 1: poste de tri

Vue détaillée – avant-projet mécanique

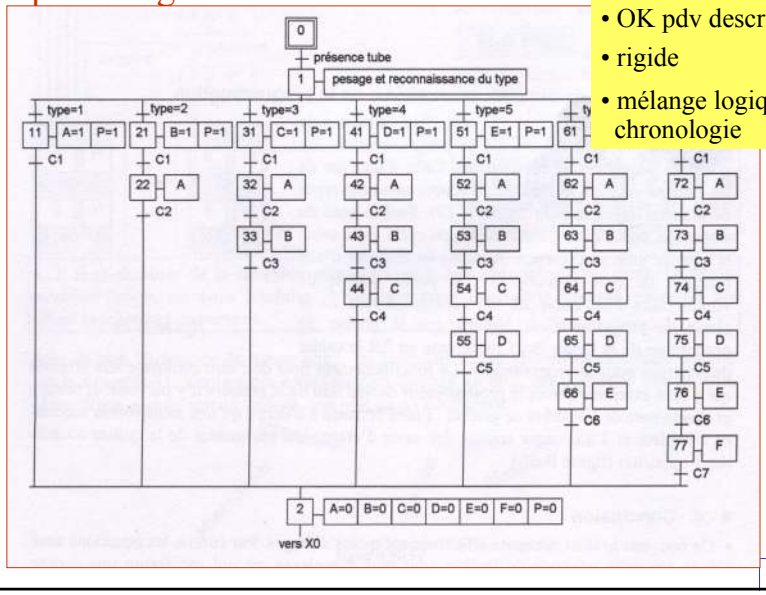


- Les détecteurs capacitifs C1 … C7 sont sensibles au passage des tubes mais ne réagissent pas lorsqu'un clapet se soulève (portée réduite des Ci)
- Les vérins (A, B, C, D, E, …) sont tous pilotés par des distributeurs monostables
- La reconnaissance des types de tubes se fait par leur pesage respectif (à l'entrée)

Problème 1: poste de tri

Un premier grafcet « à l'intution »

- Intuitif
- OK pdv descriptif
- rigide
- mélange logique et chronologie

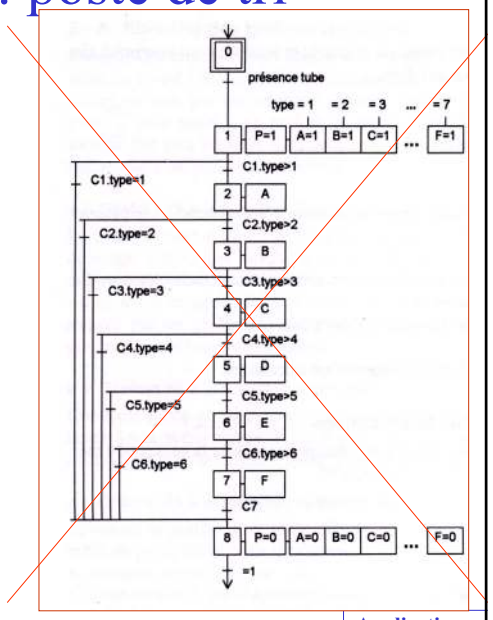


Problème 1: poste de tri

Simplification en vue de la programmation

Le grafcet précédent est donné au programmeur

- Celui-ci décide de la façon de l'implémenter
- Idée : n'écrire qu'une fois la séquence et aménager les sauts d'étapes aux moments opportuns.
  - plus concis
  - KO pdv descriptif
  - moins rigide mais reste intuitif



Problème 1: poste de tri

Réalisation méthodique du grafcet sur base de l'analyse fonctionnelle

- Ne tracer aucune étape grafcet avant d'avoir analysé les fonctions à remplir par le système
- Faire l'inventaire des fonctions / sous-fonctions
  en utilsant un vocabulaire qui fait abstraction des organes
- Eviter dans un premier temps toute référence à la chronologie du cycle .. Elle nécessite un effort « intellectuel » important

Fonctions :

transporter

alimenter

sélectionner

identifier

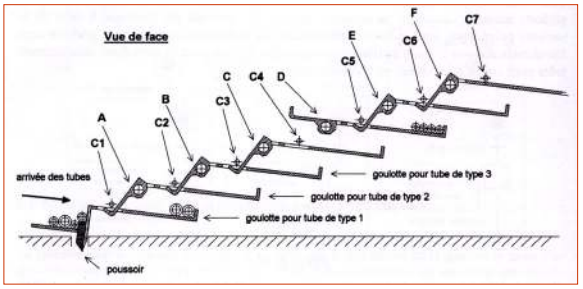
ex. alimenter = le tube présent est transféré à la première goulotte.
ex. : transporter : transporter le tube depuis son arrivée en goulotte 1 jusqu'à destination

(+ Performances : vitesse de déplacement, courses (=> conception mécanique))

Problème 1: poste de tri

Coordination des fonctions

= Quid de la chronologie des fonctions précitées (demande réflexion et décision)

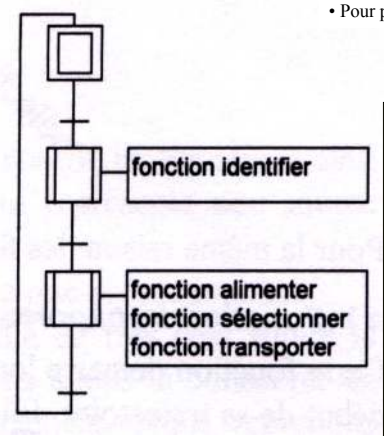


- on doit d'abord identifier (c'est une évidence)
- sélectionner peut démarrer en même temps que transporter
- alimenter doit être appelée au plus tôt (càd « pas avant ») en même temps que sélectionner pour permettre la bonne alimentation de la goulotte 1
- identifier (à nouveau) nécessite la fin de toutes les autres fonctions

Problème 1: poste de tri

Coordination des fonctions => grafcet de coordination

- on doit d'abord identifier
- sélectionner peut démarrer en même temps que transporter
- alimenter doit être appelée au plus tôt en même temps que sélectionner
- Pour permettre la bonne alimentation de la goulotte 1



Fonction principale :

« Transférer les tubes de type « i » vers l'emplacement « i »

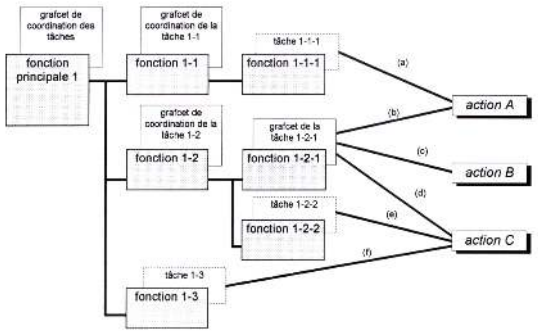
Grafcet de coordination des tâches :

- tâche (⇔ sous-fonction) « identifier »
- tâche (⇔ sous-fonction) « alimenter »
- tâche (⇔ sous-fonction) « sélectionner »
- tâche (⇔ sous-fonction) « transporter »

Problème 1: poste de tri

Grafcet lié à chaque fonction

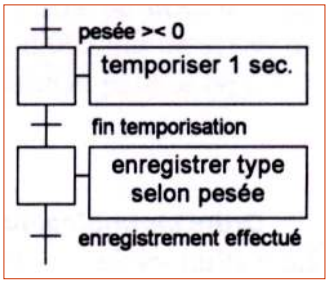
Rappel …



Suivant la complexité, il sera directement dans le grafcet de coordination ou il fera l'objet d'un Grafcet séparé, appelé via une Macro-étape du grafcet de coordination des tâches

Problème 1: poste de tri

Grafcet lié à la fonction « identifier »

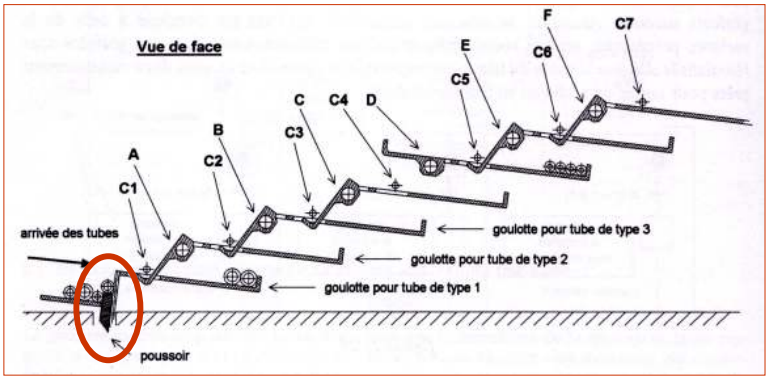


Peu de commentaire :

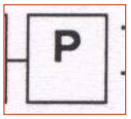
- la chronologie est évidente
- Attention : « peser » est une réceptivité - « enregistrer la pesée » est une action !
- Il faut (généralement) une tempo pour stabiliser la valeur de la pesée.

Problème 1: poste de tri

Grafcet lié à la fonction « alimenter »



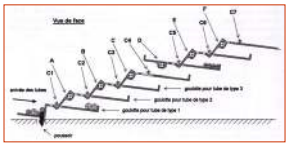
Elementaire = une action : alimenter l'électronvanne « P » du poussoir d'entrée



=> Dans le grafcet de coordination des tâches

Problème 1: poste de tri

Grafcet lié à la fonction « sélectionner »



Séquence de réflexion :

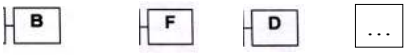
1. Inventaire des sous-fonctions et des [actions]

Sélectionner la goulotte 3 [soulever le clapet C]

Sélectionner la goulotte 2 [soulever le clapet B]

Sélectionner la goulotte 6 [soulever le clapet F]

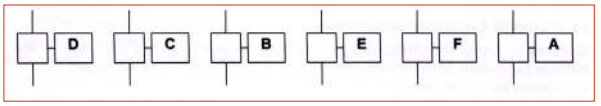
sélectionner la goulotte 7 [ne rien faire]



2. Quelle est la chronologie des actions

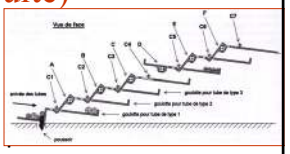
aucune ! (pour les actions précédentes !)

et donc:



Problème 1: poste de tri

Grafcet lié à la fonction « sélectionner » (suite)



Séquence de réflexion (suite)

3. Informations qui provoquent ces actions

La sélection du clapet est liée au type de tube en présence :

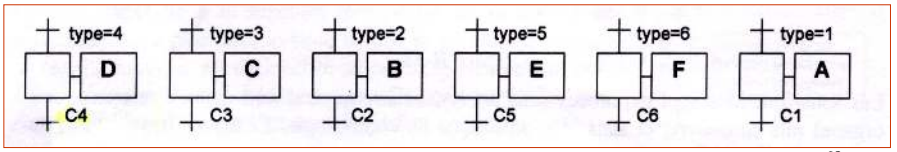
ex.: réceptivité « Type = 4 » pour l'étape associée à l'action « D »

4. Informations qui permettent l'arrêt de ces actions

Un clapet s'abaisse dès que le tube à trier est passé au dessous de celui-ci

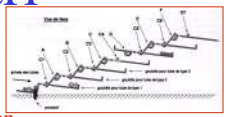
ex.: réceptivité « C4 » pour l'étape associée à l'action « D »

et donc:



Problème 1: poste de tri

Grafcet lié à la fonction « transporter »



Rappel : transporter le tube depuis son arrivée en goulotte 1 jusqu'à destination

⇒ Recherche de sous-fonctions indépendantes de la chronologie d'exécution des actions sous-jacentes (A, B, C, D, …)

~~Exemple~~
~~Sous-fonction 1 : transporter un tube de type 1~~
~~Sous-fonction 2 : transporter un tube de type 2~~
~~…~~

se réfère à la chronologie des actions

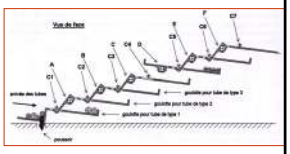
Mais :
Sous-fonction 1 : transporter un tube de la goulotte 1 à la 2
Sous-fonction 2 : transporter un tube de la goulotte 2 à la 3
…

OK !

Sous-fonctions = transports élémentaires, sans dépendance chronologique

Problème 1: poste de tri

Grafcet lié à la fonction « transporter » …



Séquence de réflexion :

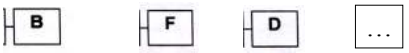
1. Inventaire des sous-fonctions [actions]

Transporter un tube de la goulotte 2 vers la 3 [soulever le clapet B]

Transporter un tube de la goulotte 1 vers la 2 [soulever le clapet A]

Transporter un tube de la goulotte 5 vers la 6 [soulever le clapet E]

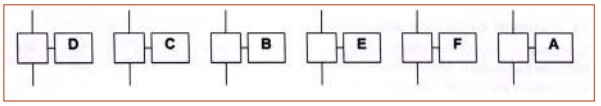
Transporter un tube de la goulotte 6 vers la 7 [soulever le clapet F]



2. Quelle est la chronologie des actions

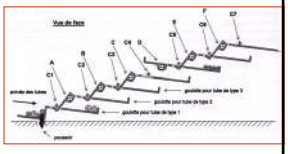
aucune car les sous-fonctions précédentes sont isolées les unes des autres.

et donc:



Problème 1: poste de tri

Grafcet lié à la fonction « transporter » (fin)



Séquence de réflexion (suite)

3. Informations qui provoquent ces actions

Il faut la présence du tube (détecteur Ci) et d'un bon tube (type > i) :

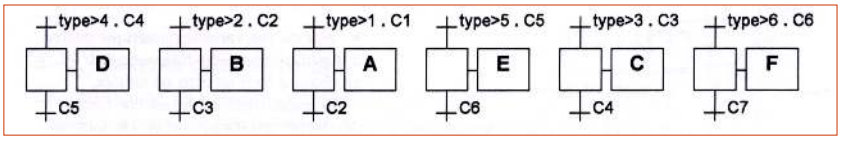
ex.: réceptivité « Type >4 . C4 » pour l'étape associée à l'action « D »

4. Informations qui permettent l'arrêt de ces actions

Un clapet peut s'abaisser lorsque le tube est dans la goulotte supérieure

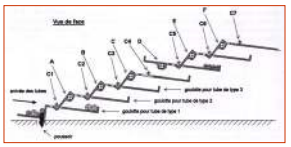
ex.: réceptivité « C5 » pour l'étape associée à l'action « D »

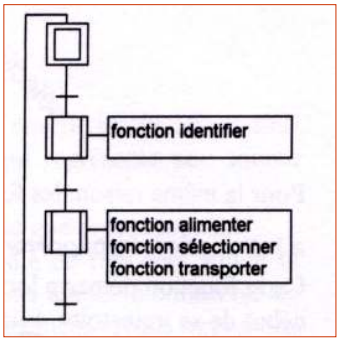
et donc:

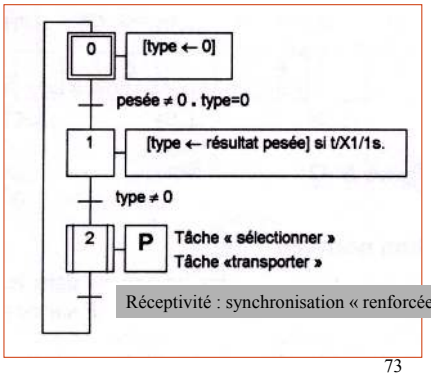


Problème 1: poste de tri

Grafcet final
(mode de production normale)



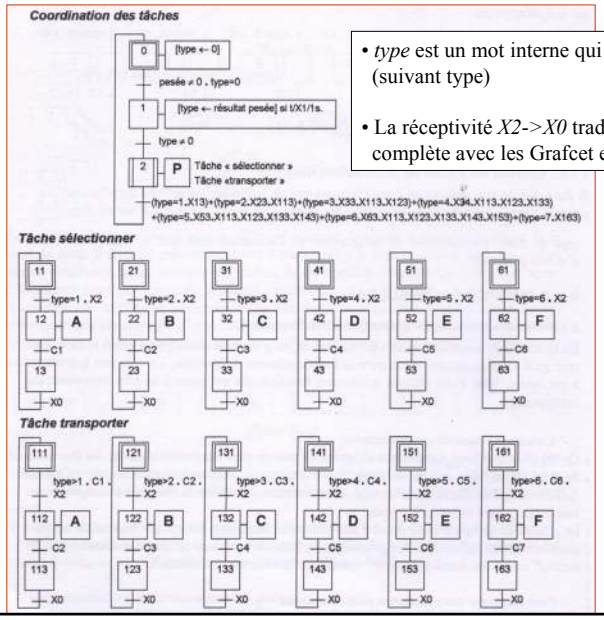




Grafcet de coordination
Des tâches (ou des fonctions)

Problème 1: poste de tri

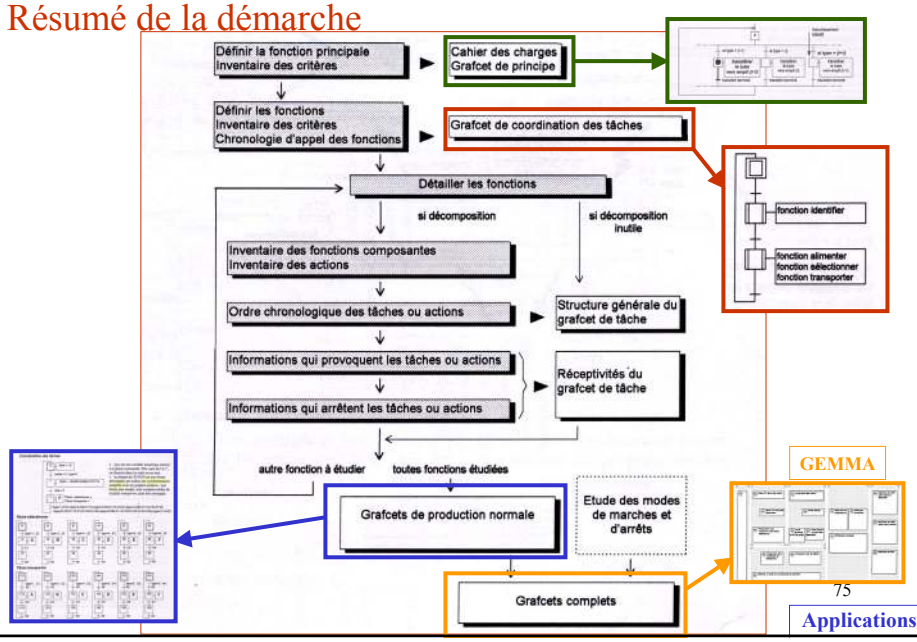
Grafcet final (mode de production normale)



- *type* est un mot interne qui varie de 0 à 7 (suivant type)
- La réceptivité *X2->X0* traduit une synchronisation complète avec les Grafcet esclaves (peut être simplifiée)

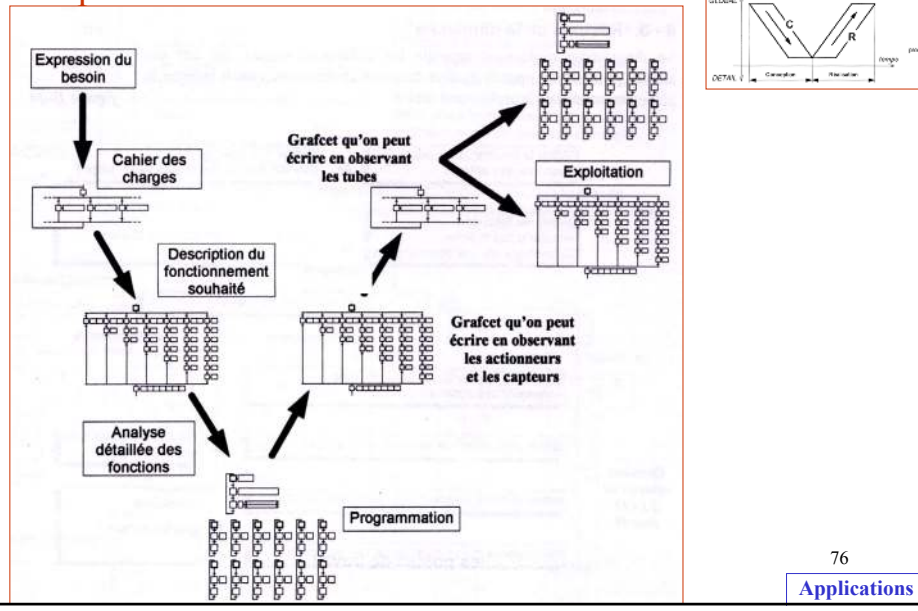
Problème 1: poste de tri

Résumé de la démarche



Problème 1: poste de tri

Récapitulation

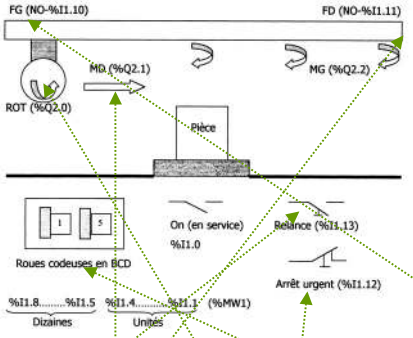


Convoyeur/Scara Laboratoire



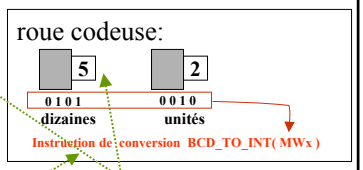
Problème 2: poste d'usinage

Enoncé



%MW0 = pointeur
%MW1 = consigne BCD (8 bits)
%MW2 = copie
%M1 = conditions initiales
%TM0 = tempo on delay

FG : fin de course gauche
FD : fin de course droit
M0 : mémoire d'arrêt d'urgence



Le commutateur « en service » lance la translation et la rotation de l'outil, pourvu que les conditions initiales soient vérifiées (%M1=1) ; l 'outil effectue des « va-et-vient » en permanence de gauche à droite.

L'amplitude du mvt de translation de l'outil vers la droite est ajustable par des « roues codeuses » (codage BCD) présentes sur le pupitre. Cette consigne représentera la présélection (%TM0.P) d 'un timer (%TM0) en secondes. Sachant que le temps nécessaire à l 'outil pour actionner le fin de course droit est de 20 sec. (=amplitude max), lorsque l'opérateur règle une présélection inférieure (0<présel<20), l'outil pourra effectuer des courses plus courtes. Le programme n'autorisera le changement de consigne (roues codeuses) qu'à l'arrêt et en position initiale (à gauche).

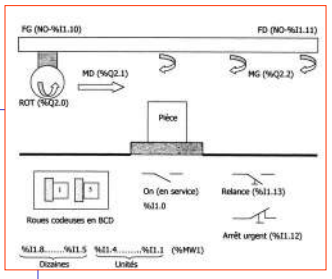
Traitement arrêt urgent et reprise:

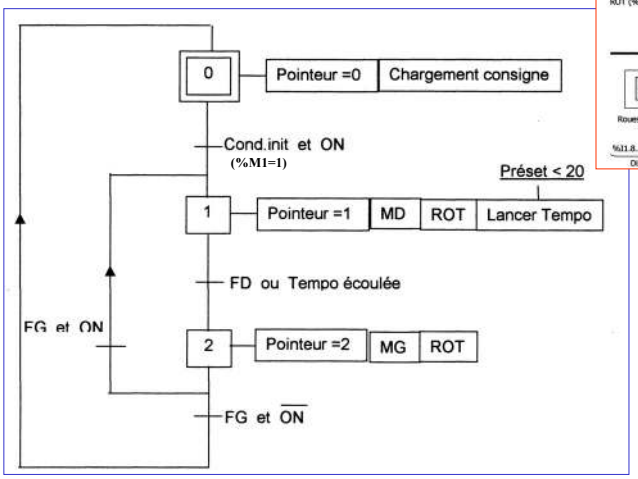
- En cas d 'incident, l'opérateur donne une impulsion sur le bouton %I1.12 qui désactive toutes les étapes du graphe, donc toutes les sorties.
- La reprise après arrêt urgent se fait à l'étape qui était active (mémorisée dans %MW0) au moment de l'arrêt: le grafcet reprend son évolution.

Problème 2: poste d'usinage

Grafcet

« ok » pour discussion
entre techniciens et ingénieurs

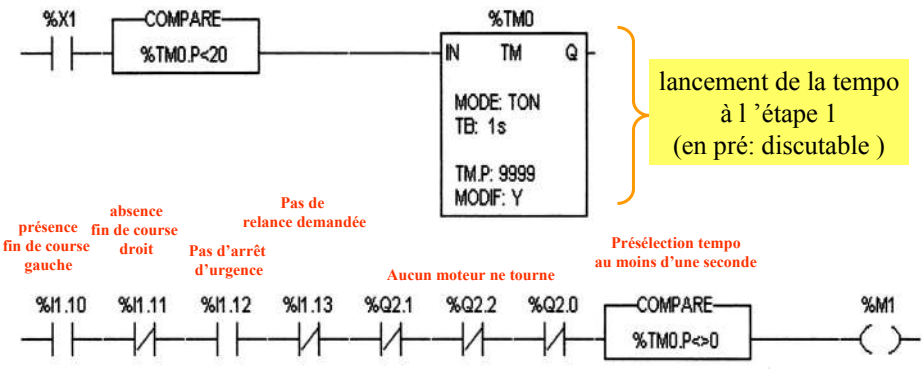




Problème 2: poste d'usinage

Actions « Pré » (1)

Lancement du Timer à l'étape 1 (*) et Test de toutes les conditions initiales



Typiquement dans le « pré » même si aurait pu être mis à l 'étape 0

(*) aurait pu se programmer dans le chart, étape 1

Problème 2: poste d'usinage

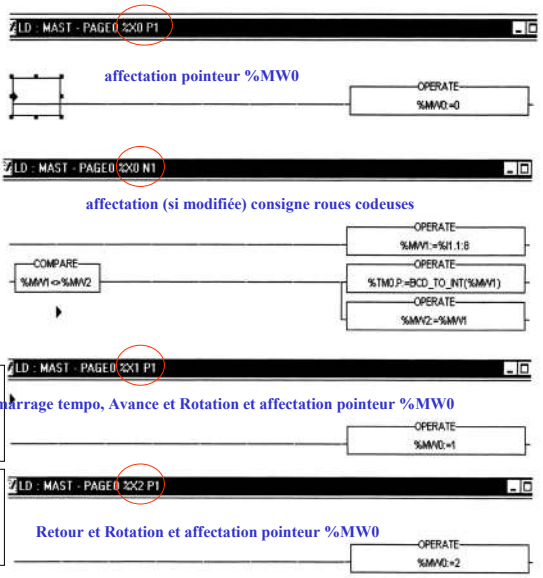
Actions « chart »

Etape 0

Transitions « chart »:
voir grafcet

Etape 1

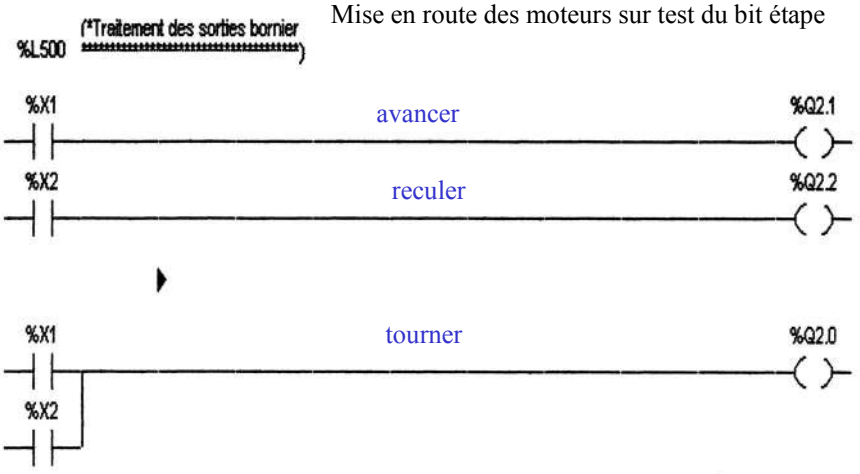
Etape 2



Problème 2: poste d'usinage

Actions « Post »

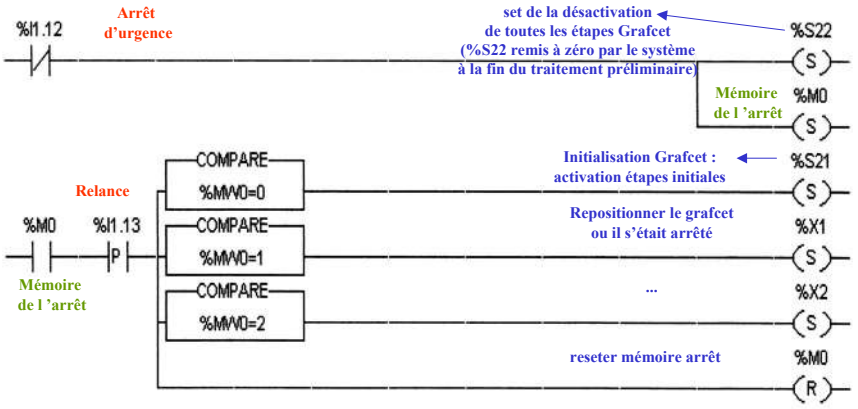
Mise en route des moteurs sur test du bit étape



Problème 2: poste d'usinage

Actions « Pré » (2)

Traitement de l'arrêt d'urgence (%I1.12) et de la reprise



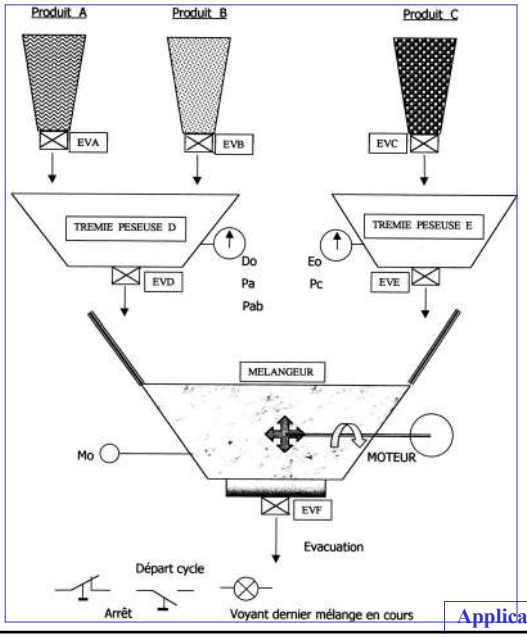
NB: %S22 = désactivation (set) des étapes grafcet ; %S21: initialisation grafcet

ça, c 'est du « vrai Pré »

Problème 3: doseur/mélangeur

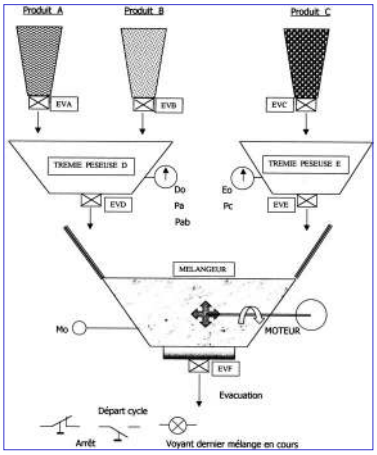
Enoncé

Soit un process qui doit:
1. doser 3 produits A, B, C
   (A et B dosés ensemble :
   A puis A+B,
   C, dosé séparément)
2. mélanger les trois produits
3. évacuer les produits



Problème 3: doseur/mélangeur

Capteurs/Actionneurs



Do : %M8 : détecteur de trémie D vide (NO).

Pa : %M9 : détecteur de dosage produit A terminé (NO).

Pab : %M10 : détecteur de dosage global A et B terminé (NO).

Eo : %M11 : détecteur de trémie E vide (NO).

Pc : %M12 : détecteur de dosage produit C terminé (NO).

Mo : %M13 : détecteur d'absence de matière mélangeur (NO).

Dcy : %I1.14 : BP (NO) départ cycle.

$\overline{A}$: %I1.15 : BP (NF) arrêt de production.

EVA : %Q2.8 : électro-vanne monostable d'ouverture du silo A.

EVB : %Q2.9 : électro-vanne monostable d'ouverture du silo B.

EVC : %Q2.10 : électro-vanne monostable d'ouverture du silo C.

EVD : %Q2.11 : électro-vanne monostable d'ouverture trémie D.

EVE : %Q2.12 : électro-vanne monostable d'ouverture trémie E.

EVF : %Q2.13 : électro-vanne monostable d'ouverture mélangeur.

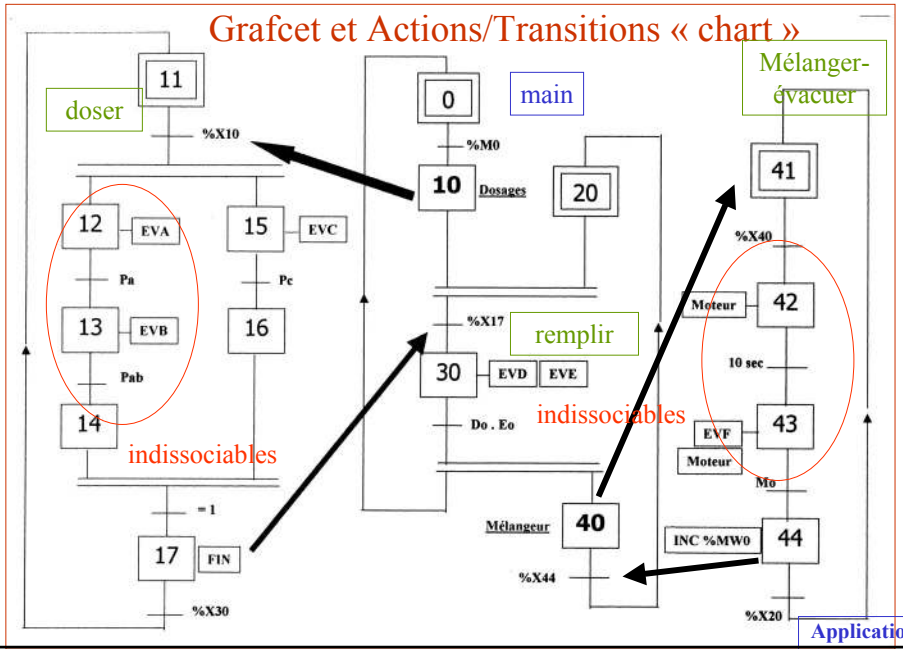
M : %Q2.14 : contacteur du moteur du mélangeur.

V : %Q2.15 : contacteur du voyant clignotant dernier mélange en cours.

%MW0 : nombre de mélanges produits.

Problème 3: doseur/mélangeur

Grafcet et Actions/Transitions « chart »

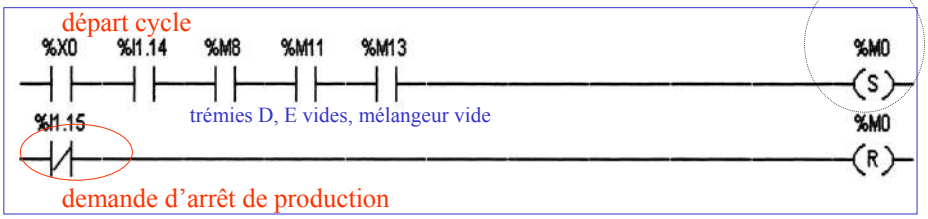


Problème 3: doseur/mélangeur

Actions « Pré »

Gestion de l'arrêt

condition démarrage cycle (voir grafcet)

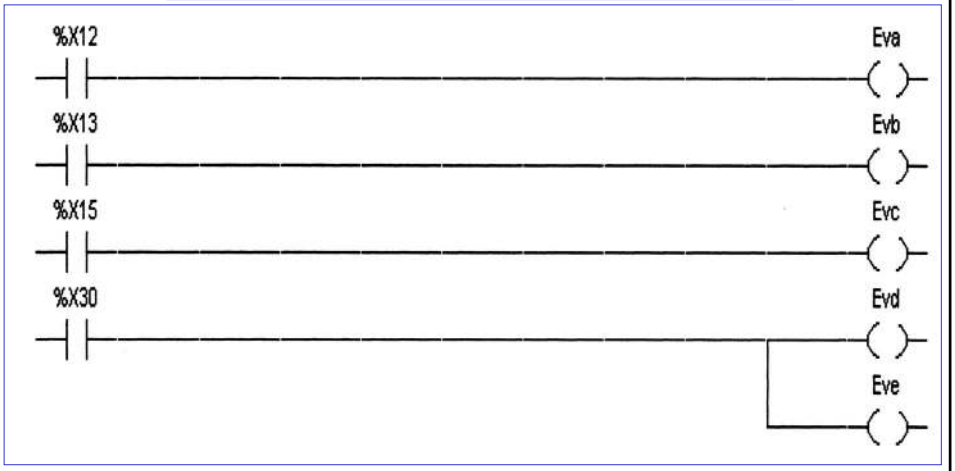


en « pré » car c'est de la gestion « marche/arrêt »

Problème 3: doseur/mélangeur

Actions « Post » (1)

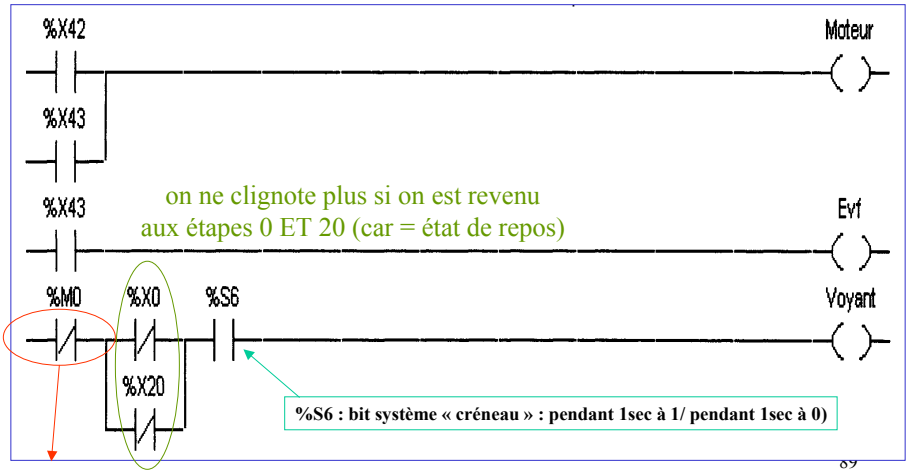
Gestion des dosages A,B,C et évaculation trémies D et E



Problème 3: doseur/mélangeur

Actions « Post » (2)

Gestion du mélangeur: moteur et évacuation
Gestion du clignotant (dernier mélange en cours)



si il y a une demande d'arrêt de production