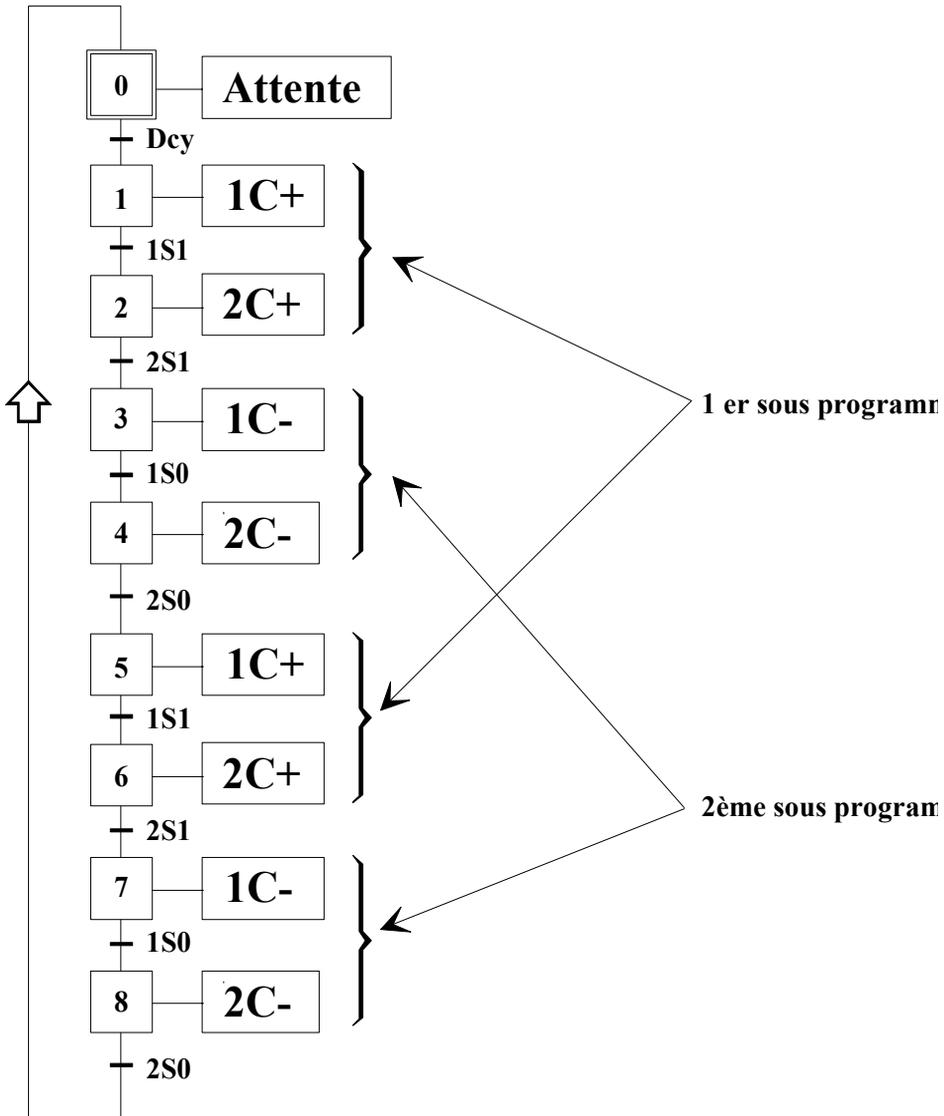


# NOTION DE SOUS PROGRAMME

S'il s'agit d'un sous programme tous les grafecets sont de même niveau hiérarchique.

## I. EXEMPLE:

### A. Point de vue P.O.



Bien que cet exemple trop simple ne fasse pas ressortir l'intérêt d'une telle écriture les sous programmes ne font que 2 étapes

On s'aperçoit que si ceux-ci faisaient 30 étapes ou plus avec 5 ou 10 reprises dans le même programme On aurait un grafset difficilement compréhensible et disproportionné, d'autre part si quelques paramètres changent on doit tout refaire, alors que cette méthode de sous programme nous évite tous ces problèmes.

## II. PRINCIPE:

Nous allons considérer que:

- ⇒ la première moitié de cycle 1C+, 2C+ est un sous programme
- ⇒ la deuxième moitié de cycle 1C-, 2C- est un autre sous programme

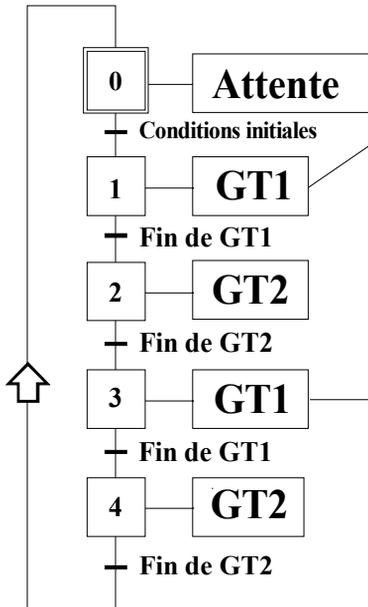
Un sous programme est une tâche nous appellerons ces 2 grafset GT1 (Grafset de Tâche N°1) et GT2

Un grafset de coordination des tâches donnera l'ordre d'enclenchement de ces deux sous programmes.

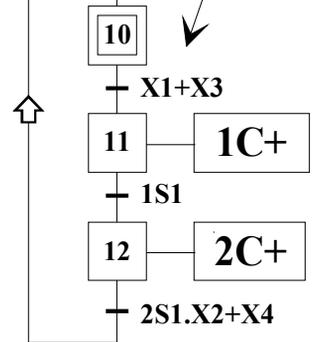
La flèche indique la manière dont on se branche dans le sous programme.

Compléter par des flèches de couleurs branchements et réceptivités

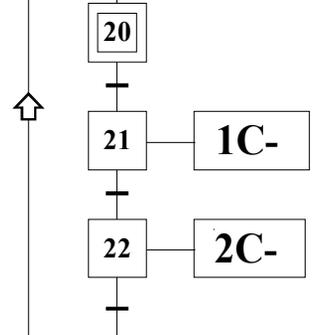
### Grafset de coordination



### GT1



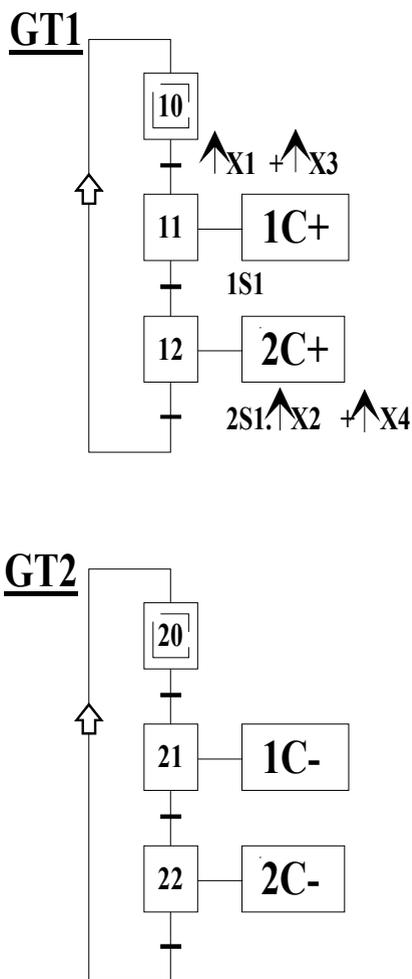
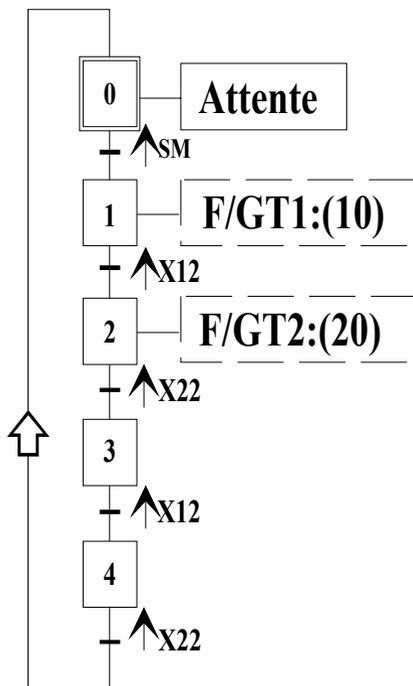
### GT2



Il n'est pas toujours souhaitable à la mise sous tension que le bit étape des sous programmes soit à 1, ce qui dans certains cas permettrait un démarrage intempestif et non souhaité . pour être sur du fonctionnement on procède par forçage d'étape ( un grafcet ne peut pas démarrer sans étape initiale sauf si une étape est forcée à 1, l'inverse du forçage est le figeage qui est un forçage à 0.

## 1 POINT DE VUE P.C.

### Grafcet de coordination



Une fois forcée les étapes 10 et 20 le reste (étape 1 et 2 du grafcet de coordination), c'est pourquoi il n'est pas utile de les renforcer.

L'étape en pointillée signifie ordre interne à l'automate

Le cadre en pointillé GT1 et GT2 signifie étape forçage à partir d'un autre grafcet

## 2 PARTIES COMMANDES DECENTRALISEES

- De plus en plus les automatismes se décentralisent.
- On peut avoir pour chaque poste ou pour chaque tâche une sous-partie commande.
- Une des sous-partie commande peut être maitre ou les autres esclaves.

- Où toutes les sous parties commandes sont au même niveau.

### **3 REGLES D'EVOLUTION POUR LE FORÇAGE:**

#### **A. Règle 1:**

- Les grafctet forcés prennent immédiatement et directement la situation imposée.

#### **B. Règle 2:**

- L'ordre de forçage est prioritaire par rapport à toute l'activité du grafctet. Les grafctets forcés sont maintenus dans la situation imposée tant que les ordres de forçage sont validés.

### **III. GRAFCETS SYNCHRONISES**

- On utilise la notion d'activité d'étape  $X_i$ ,  $i$  étant le repère de l'étape.
- Il faut faire attention à cette notation. Si elle simplifie les grafctets *elle masque*

#### ***les structures de fonctionnement.***

#### **A. Exemple:**

dans le premier cas on comprend très bien le fonctionnement du SAP . Dans le 2ème cas si l'on ne maîtrise pas le grafctet il n'est pas évident d'y voir le même fonctionnement. C'est pourquoi il ne faut pas hésiter à être redondant et expliquer le fonctionnement globale par un grafctet simplifier dans le style du premier

#### **On l'utilise :**

- entre les grafctets de coordination des tâches et les grafctets de tâches.
- entre le grafctet de conduite et les grafctets de chaque mode de marche
- quand il n'existe pas de structure (exemple parralélisme interprété.)

### **IV. DEPANNAGE:**

Il n'est pas possible de dépanner les SAP utilisant des grafctets synchronisés sans en comprendre parfaitement le fonctionnement.

S'ils sont hierarchisés il faut indentifier quel est la couche concernée par la panne.