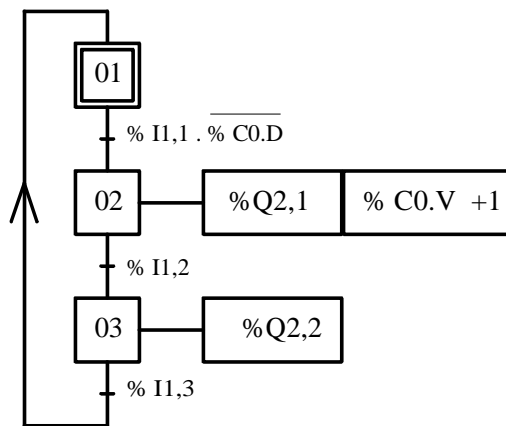


Les automatismes étant très diversifiés, on a créé des structures de grafset adaptées pour permettre leur fonctionnement optimal.

1° - Structure linéaire :

C'est le grafset le plus simple ; les étapes s'enchaînent de façon linéaire d'où une programmation simple. On l'utilisera aussi souvent que possible sur des automatismes de base mais ses limites seront vite atteintes sur des automatismes plus complexes.

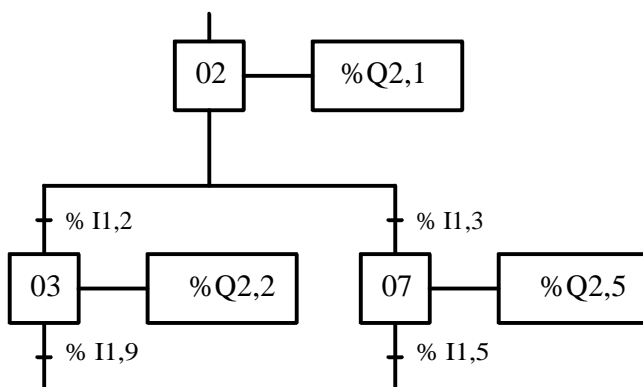
ex :



2° - Divergence et convergence en OU : choix de séquence

Cette structure est utilisée sur les automatismes où le cycle est différent selon un choix imposé soit par l'opérateur, soit par des capteurs présent sur la machine :

- par l'opérateur : commutateur : marche rapide / marche lente
- par des capteurs : pièce bonne / pièce mauvaise

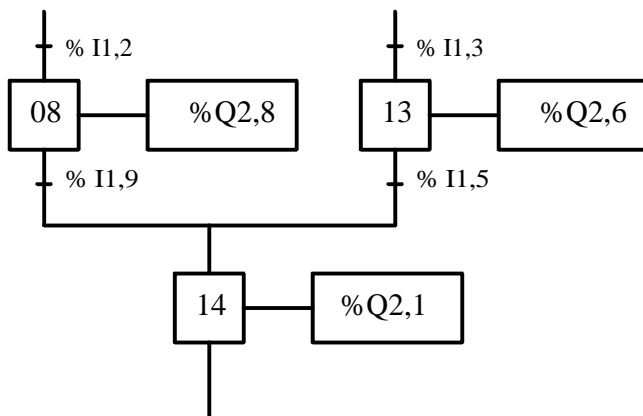


Lorsque l'étape 2 est active, on ira :

- soit vers l'étape 3 si $\%I1.2 = 1$
- soit vers l'étape 7 si $\%I1.3 = 1$

Pour éviter toute ambiguïté, on fait en sorte que les entrées $\%I1.2$ et $\%I1.3$ ne puissent pas être à 1 simultanément (ex: commutateur 2 ou 3 positions, verrouillage entre capteurs, etc...) ou on verrouille dans le programme :

$\%I1.2.\overline{\%I1.3}$ et $\overline{\%I1.2}.\%I1.3$



Selon la branche sélectionnée au départ, on ira :

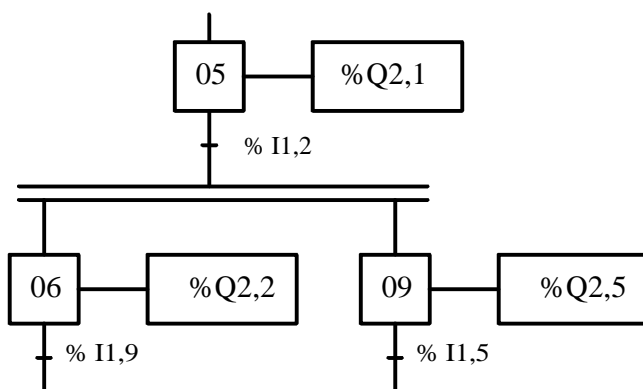
- soit de l'étape 8 vers 14
- soit de l'étape 13 vers 14

Dans les 2 cas, la transition est classique (équivalente à une structure linéaire)

* Exemple d'application : Tri de caisse (TP5)

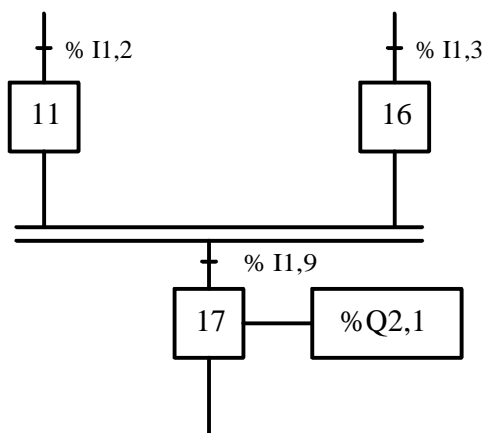
3° - Divergence et convergence en ET : Séquences simultanées

Cette structure est présente chaque fois qu'une machine comporte plusieurs postes de travail qui fonctionnent en même temps. Pour marquer la divergence en ET, on double le trait de divergence.



Lorsque l'étape 5 est active et que la réceptivité %I1.2 est vraie (=1) :

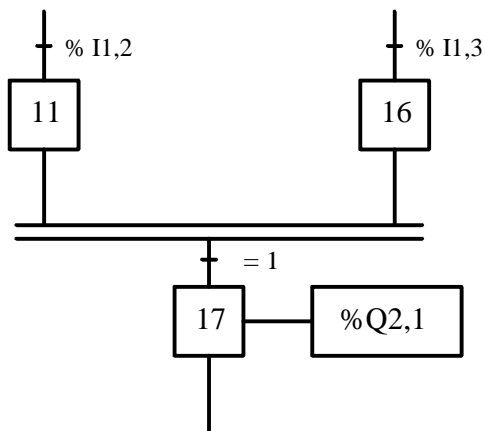
- les étapes 6 et 9 sont activées simultanément
- les 2 branches de Grafcet évoluent simultanément



Pour valider l'étape 17, il faut :

- que les étapes 11 et 16 soient actives
- que la réceptivité %I1.9 soit vraie (=1)

Dans la majorité des cas, sur une convergence en ET, la réceptivité est inutile et donc considérées toujours vraie (notée = 1) ; C'est à dire pas d'autre condition particulière de passage de la transition correspondante hors la validation des étapes précédentes.

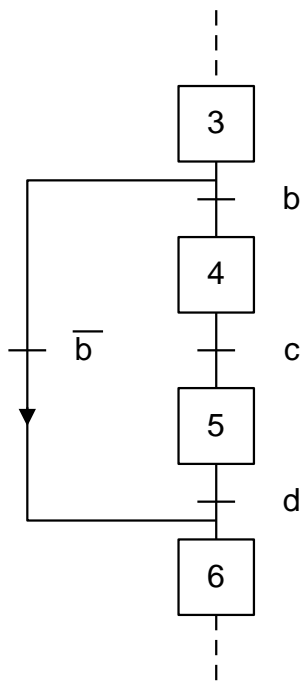


Dans ce cas, pour valider l'étape 17, il suffit que les étapes 11 et 16 soient actives.

On dit alors que ce sont des étapes de synchronisation qui n'ont pour seule utilité de synchroniser les différentes branches les unes par rapport aux autres.

* Exemple d'application : chaîne de remplissage de bidon d'huile et TP6

4° - Saut d'étape :

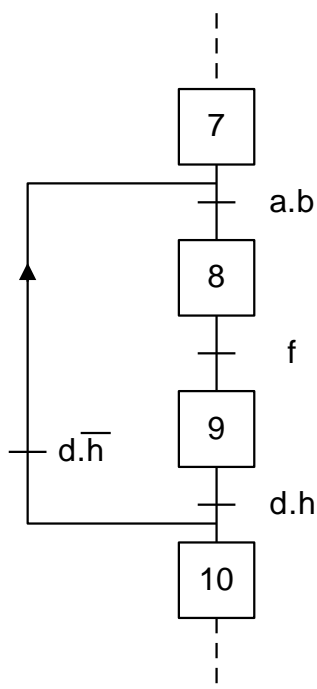


Il s'agit d'une divergence en **OU** mais qui **ne contient pas d'étape**. Cela permet d'éviter des étapes qui n'ont pas lieu de se dérouler si les conditions les concernant ne sont pas remplies.

Par exemple ici, on passera directement de l'étape 3 à l'étape 6 tant que la réceptivité entre les étapes 3 et 4 (**b**) ne sera pas vraie.

* Exemple d'application : « perceuse avec et sans débouillage »

5° - Reprise de séquence :



Si dans une application particulière, on a une suite d'actions (donc d'étapes) qui est répétitive, on peut éviter de la répéter en utilisant une reprise de séquence.

Par exemple ici, tant que la réceptivité entre l'étape 9 et l'étape 10 n'est pas vraie (**d.h**), le cycle se reboucle sur les étapes 8 et 9.

* Exemple d'application : « Traitement de surface »