



Les Automates Programmables Industriels (API)

Préparé par : Mme I. Lachkar

Historique

A la fin des années 60, Un fabricant américain de voitures décide de remplacer les systèmes de commande à base de logique câblée (relais électrique) par une logique programmée.

Ambiance industrielle

bruit électrique

poussière

température

humidité

Objectifs

Solution pour un coût acceptable

utilisable par le personnel en place

Facilement programmable

Solution qui supporte l'ambiance industrielle

Adaptable aux nombreuses variétés E/S

Mise en œuvre simple...

Juillet 1969

La Mission Apollo XI dépose N. Armstrong sur la lune ! Deux GE 635 ont contrôlé le vol.



l'ordinateur en 1969

coût "astronomique"
utilisation complexe
nécessite un environnement
particulier

Le premier PLC, modèle 084, a été inventé par Dick Morley en 1969.

Le "084" - Détails

Les "084" se composent de trois composants principaux montés sur deux rails verticaux, dont un était articulé pour permettre l'accès de service à l'avant et à l'arrière.

Ladder Logic:

L'utilisation de la Logique Ladder a été significative dans l'acceptation rapide du "084" parce que les mêmes ingénieurs et électriciens qui ont conçu et maintenu Factory Automation Systems pourraient également programmer un "084". La Logique Ladder était simplement une version électronique du schéma électrique élémentaire qu'ils utilisaient déjà - ce qui n'était pas le cas pour d'autres types de systèmes de contrôle en cours de conception.



Input/Output Rack (top)

Deux racks d'E / S peuvent être montés sur le "084" pour une capacité totale de 256 points d'E / S (un seul monté sur l'unité illustrée).

CPU (moyen)

L'unité centrale contenait le CPU. Le "084" avait 1K x 16 bits mémoire de base, qui comprenait à la fois la mémoire du système d'exploitation ainsi que le programme utilisateur.

Alimentation (basse)

Une ligne monophasée de 115 V a été connectée à l'avant du module d'alimentation, qui a fourni du courant continu au reste de l'unité.

Options:

Timer

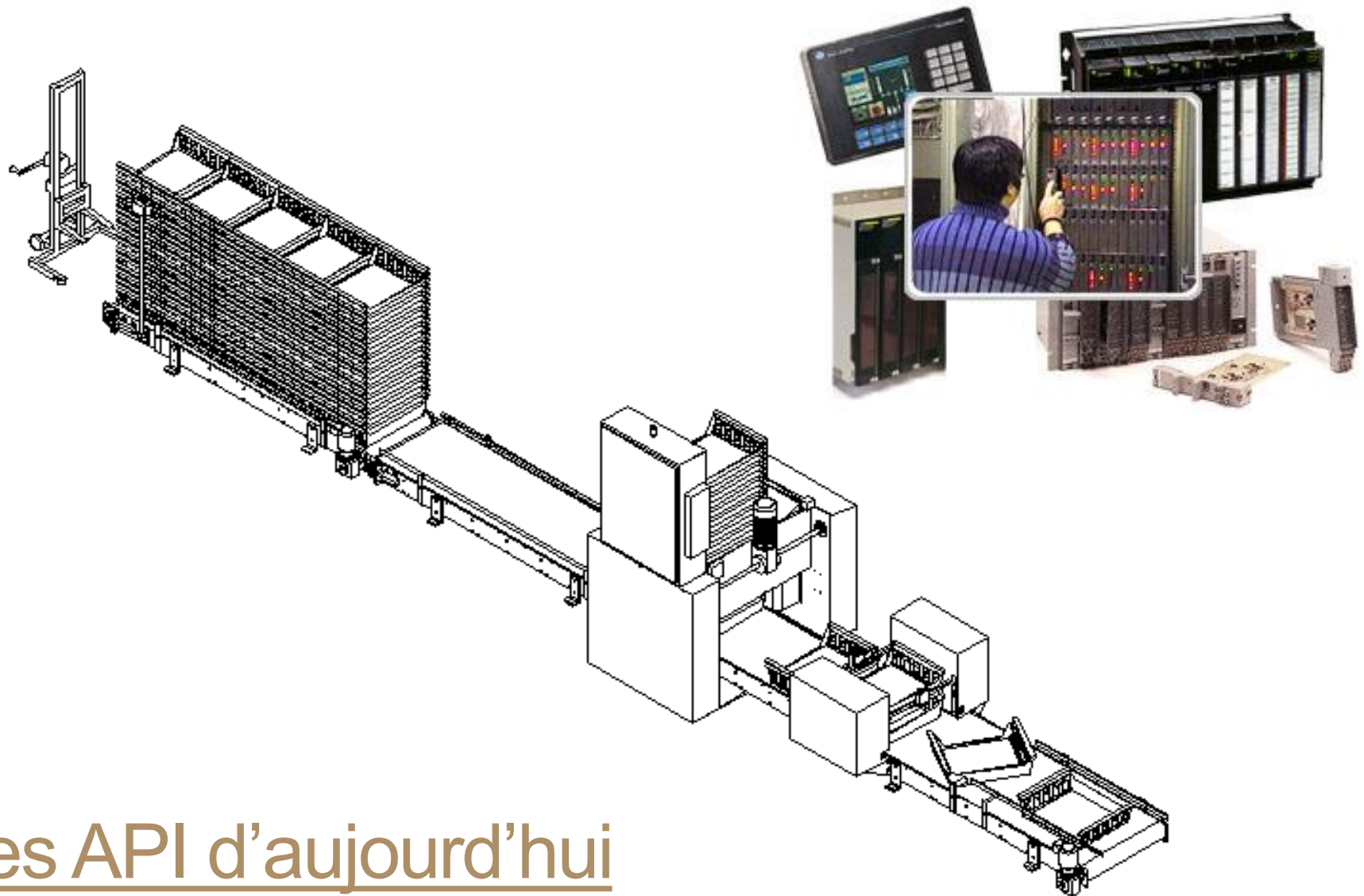
Counter

Portable Programming Panel

Magnetic Tape Cartridge Program

Loader





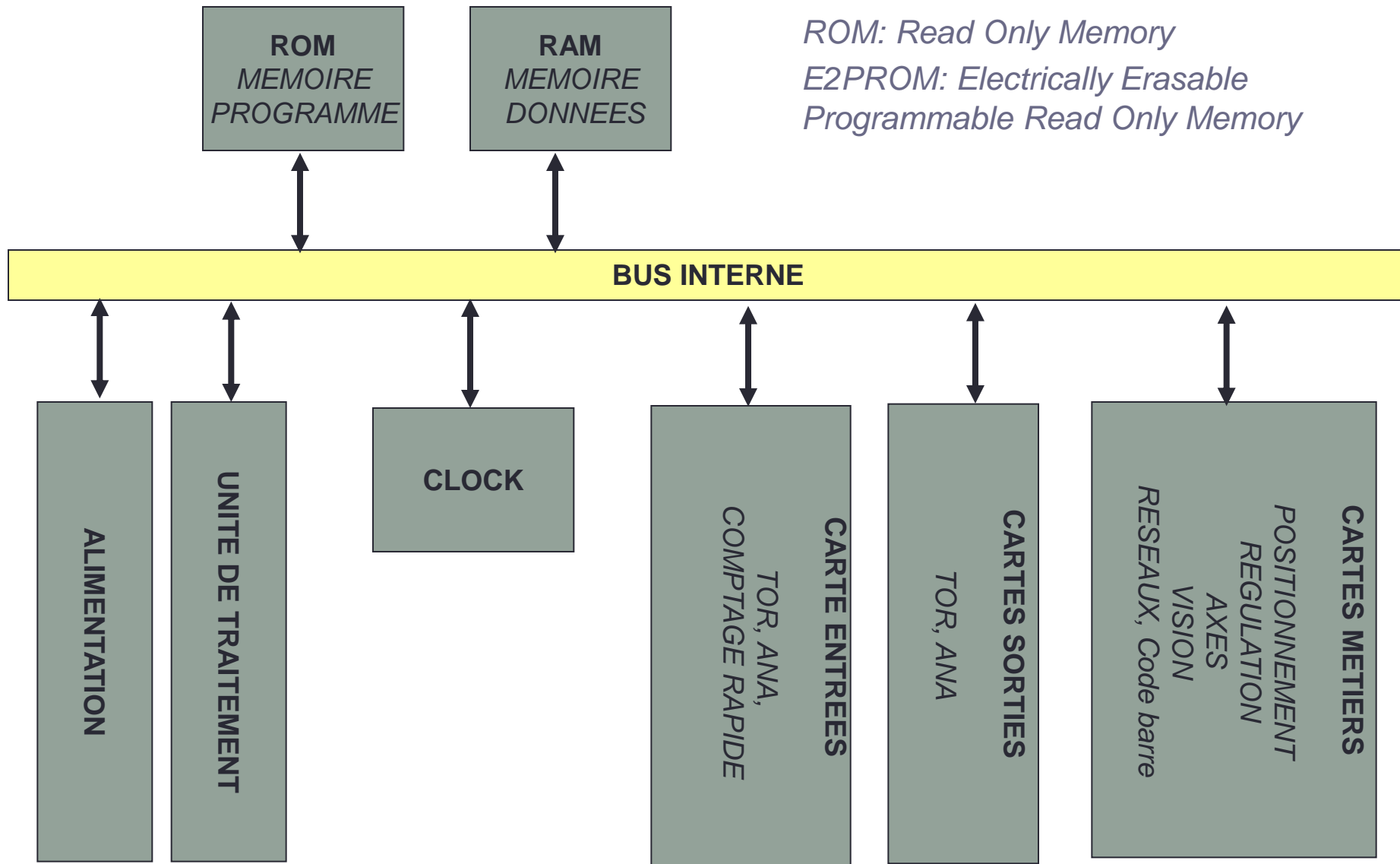
Les API d'aujourd'hui

Architecture interne

RAM: Random Access Memory

ROM: Read Only Memory

*E2PROM: Electrically Erasable
Programmable Read Only Memory*



classification

nombre
d'équipement

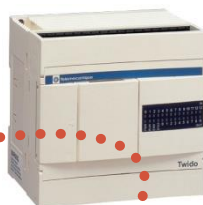
Cartes dédiées



?

SOFT PLC : Pc industriel
et logiciel de contrôle
commande

automate compact

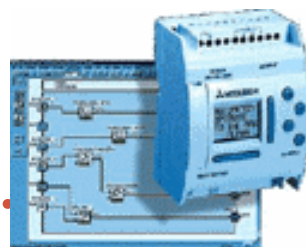


automate
modulaire

(réseaux et
métiers)



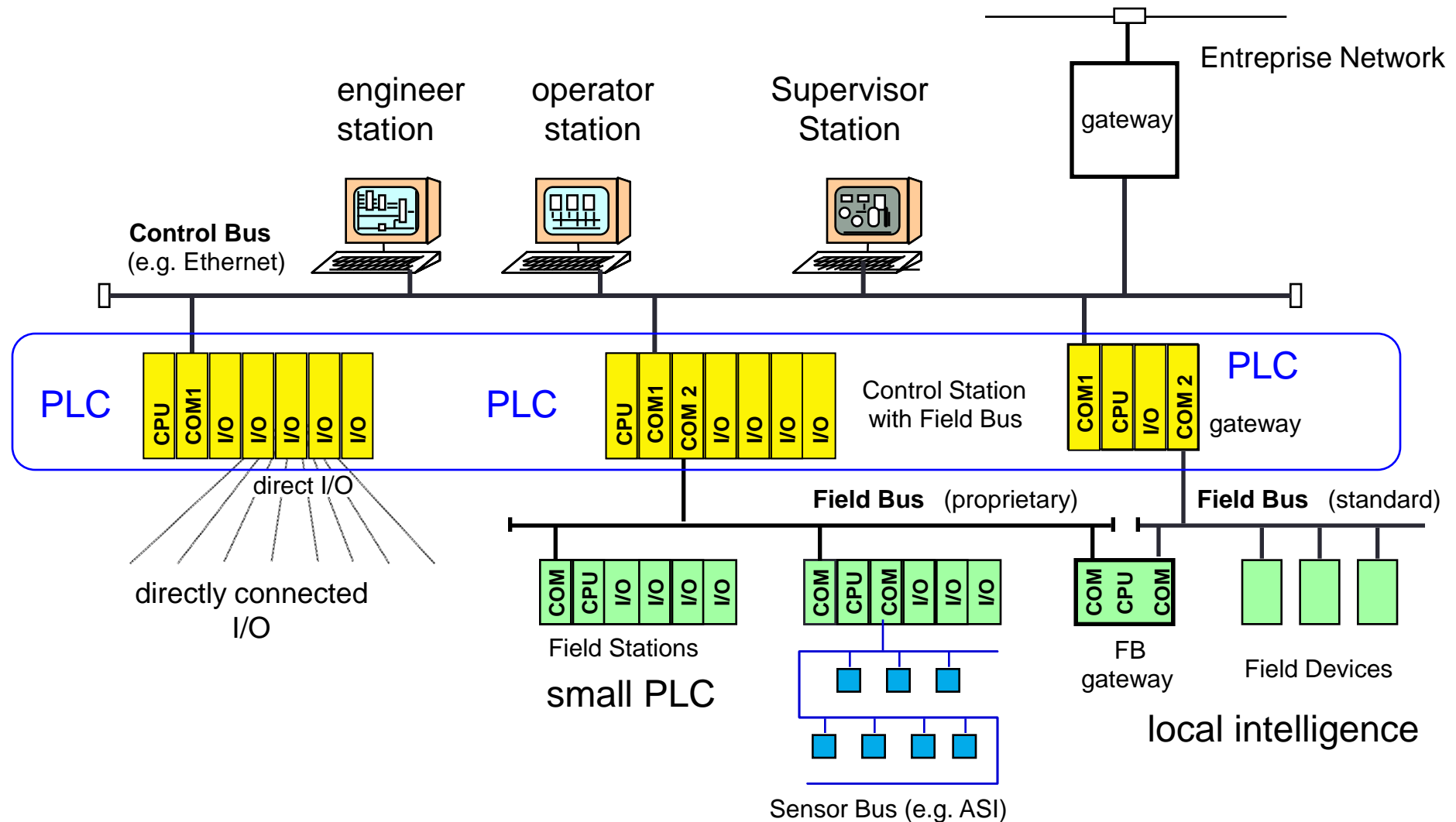
Logique
cablée

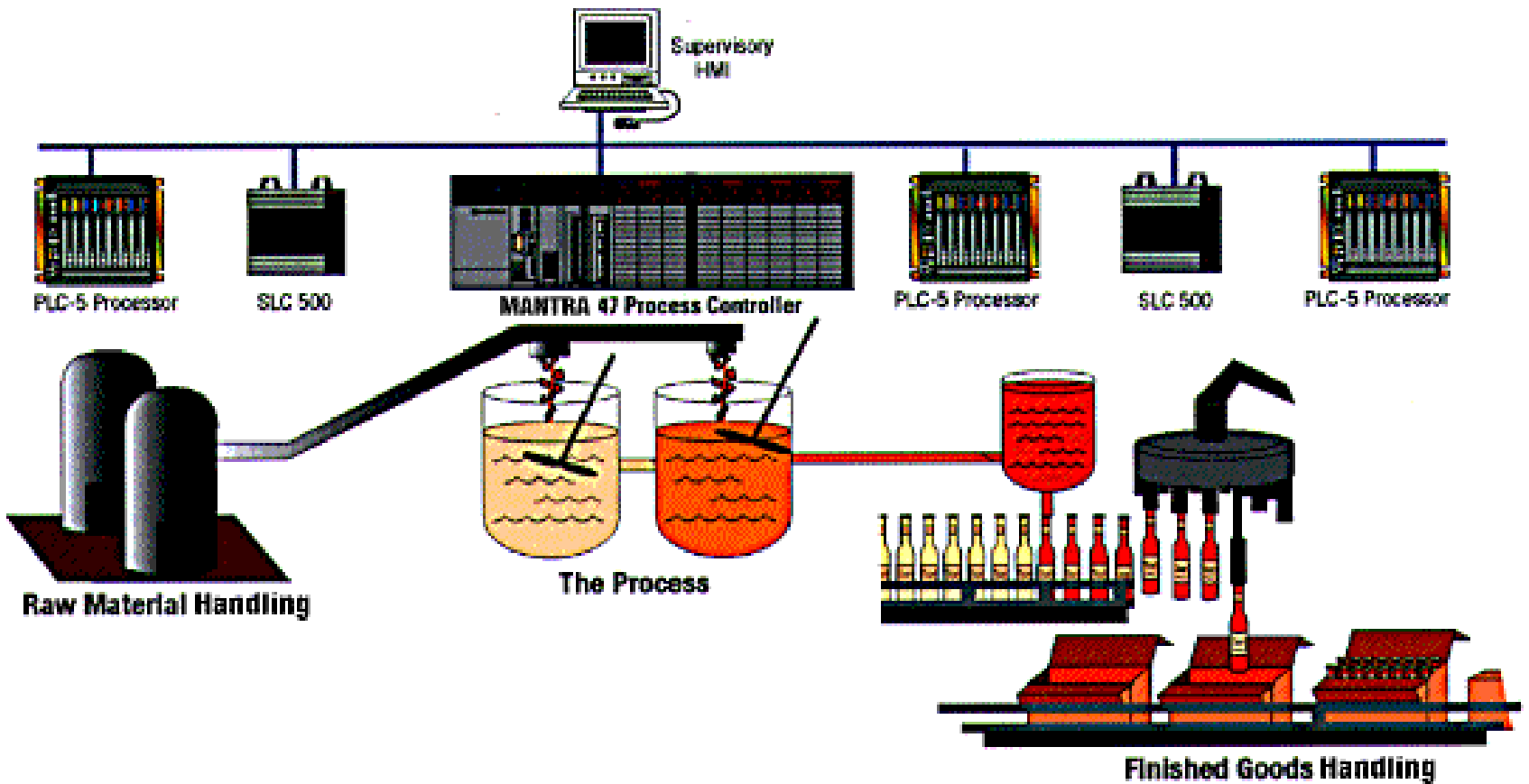


Relais
programmable

Volume & niveau
d'automatisme

Architecture réseau

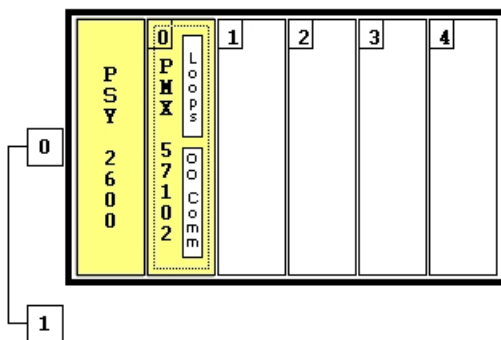
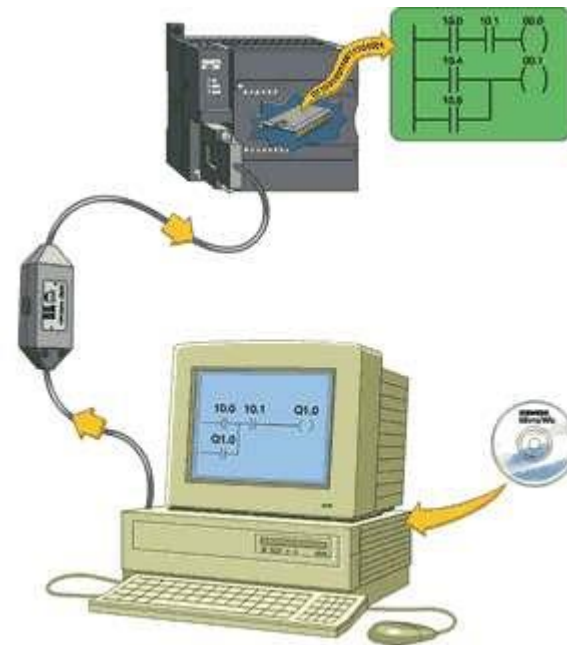




Aspect programmation

FONCTIONS DES ATELIERS LOGICIEL

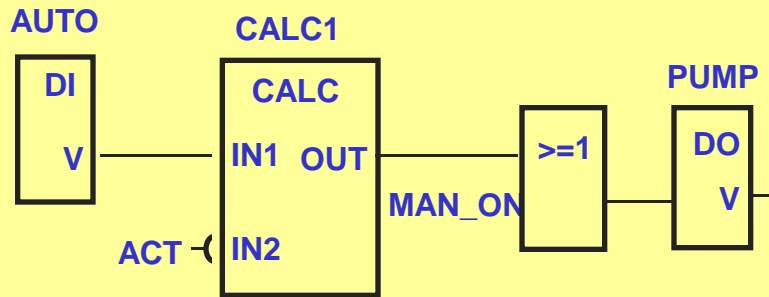
- *gestion des projets*
- *configuration API*
- *édition, compilation*
- *Transfert console <> API*
- *Mise au point dynamique*



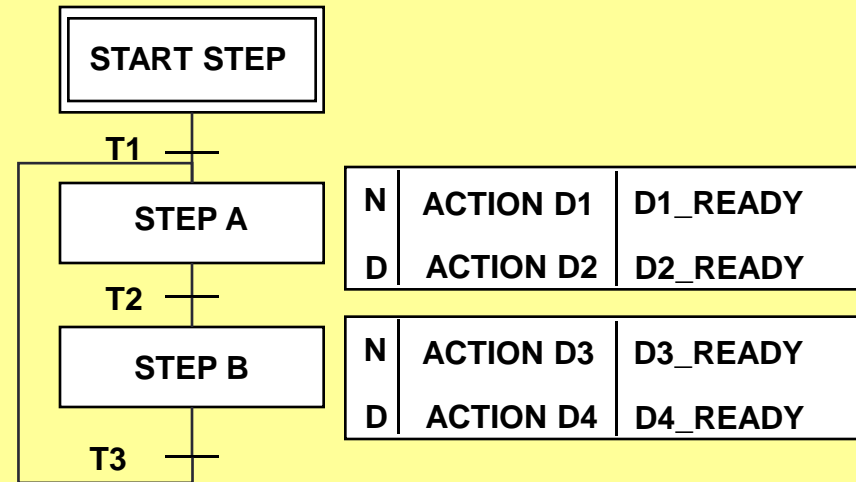
Console de poche pour la maintenance

Les langages de programmation

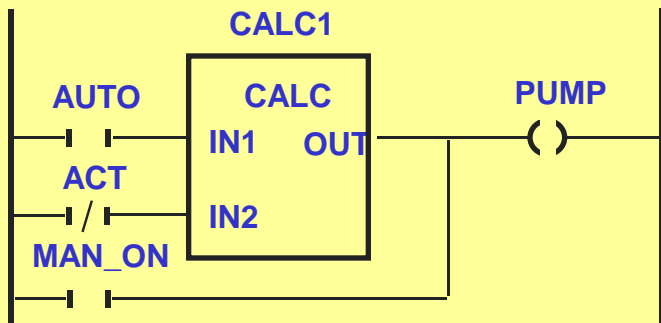
Function Block Diagram (FBD)



Sequential Flow Chart (SFC)



Ladder Diagram (LD)



Structured Text (ST)

```

VAR CONSTANT X : REAL := 53.8 ;
Z : REAL; END_VAR
VAR aFB, bFB : FB_type; END_VAR

bFB(A:=1, B:='OK');
Z := X - INT_TO_REAL (bFB.OUT1);
IF Z>57.0 THEN aFB(A:=0, B:="ERR");
ELSE aFB(A:=1, B:="Z is OK");
END_IF

```

Instruction List (IL)

```

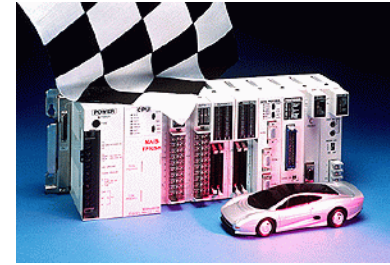
A: LD    %IX1  (* PUSH BUTTON *)
ANDN   %MX5  (* NOT INHIBITED *)
ST     %QX2  (* FAN ON *)

```

Comparaison des langages

LANGAGE	AVANTAGES	INCONVENIENTS
LD	facile à lire et à comprendre par la majorité des électriciens langage de base de tout PLC	suppose une programmation bien structurée
FBD	Très visuel et facile à lire	Peut devenir très lourd lorsque les équations se compliquent
ST	Langage de haut niveau (langage pascal) Pour faire de l'algorithmique	Pas toujours disponible dans les ateliers logiciels
IL	langage de base de tout PLC type assembleur	très lourd et difficile à suivre si le programme est complexe Pas visuel.
SFC	Description du fonctionnement (séquentiel) de l'automatisme. Gestion des modes de marches Pas toujours accepté dans l'industrie...	Peu flexible

Avantages des API



évolutivité	très favorable au évolution. très utilisé en reconstruction d'armoire.
fonctions	assure les fonctions Conduites, Dialogue, Communication et Sûreté.
taille des applications	gamme importante d'automate
vitesse	temps de cycle de quelque ms
modularité	haute modularité. présentation en rack

développement d'une application et documentation

très facile avec des outils de programmation de plus en plus puissant

architecture de commande

centralisée ou décentralisée avec l'apparition d'une offre importante en choix de réseaux , bus de terrain, blocs E/S déportées.

mise en oeuvre

mise au point rendu plus facile avec l'apparition des outils de simulation de PO

maintenance

échange standards et aide au diagnostique intégrée

portabilité d'une application

norme IEC 1131

Les fabricants renommés..

www.schneider-electric.com

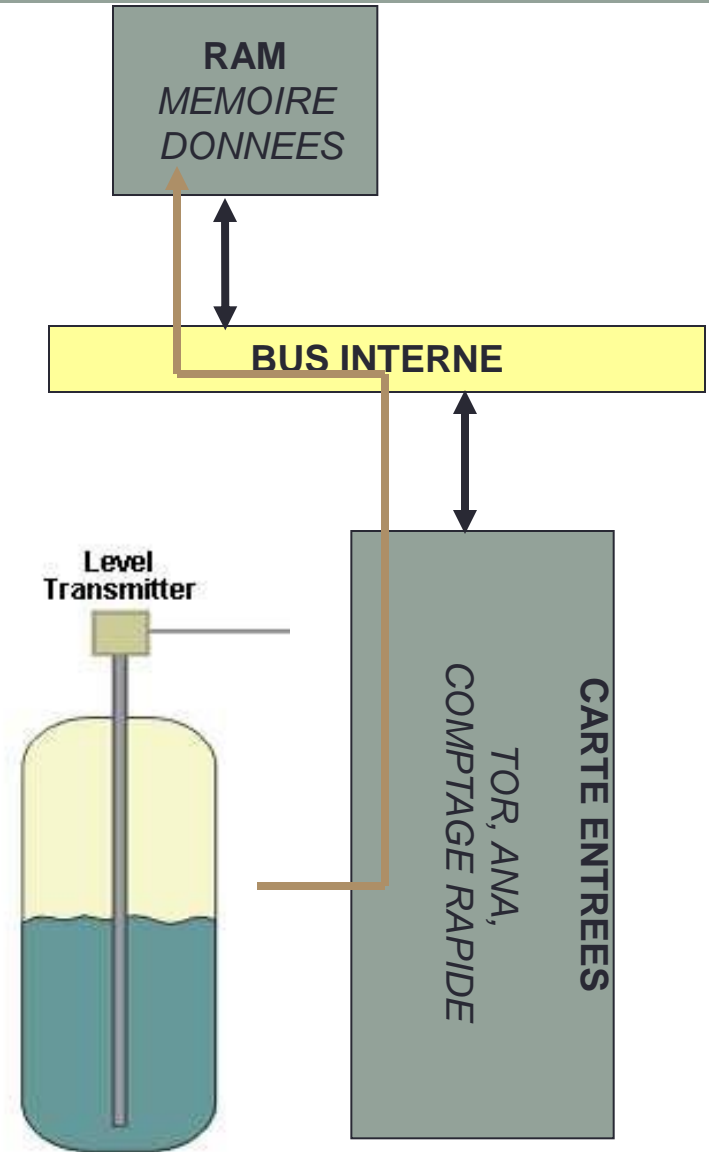


Traitement : définitions

Acquisition
des entrées

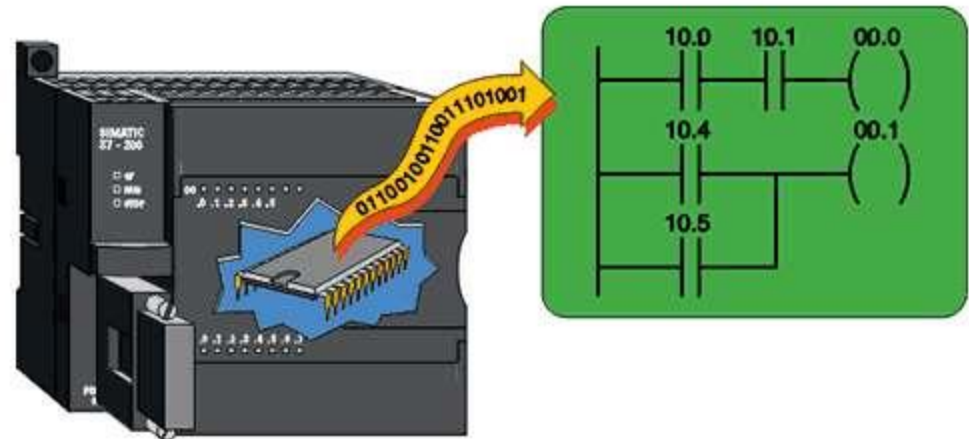
E

écriture en mémoire de
l'état des informations
présentes sur les entrées
(réalise une image du
monde extérieur)



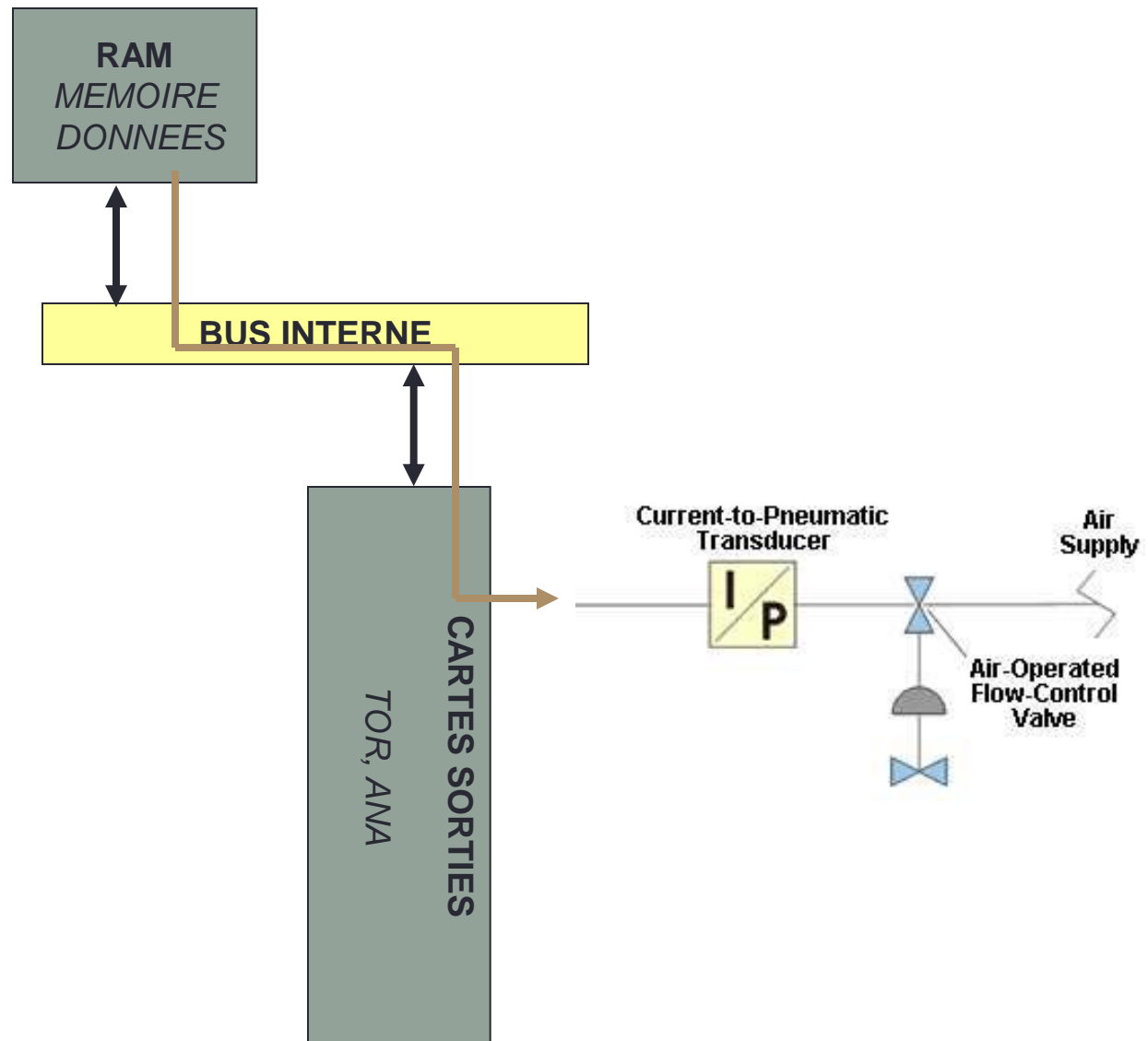
Traitement
du programme
T

exécution du
programme
application, écrit
par l'utilisateur.



Mise à jour
des sorties
S

écriture des bits ou
des mots de
sorties associés
aux modules TOR
et métier selon
l'état défini par le
programme
application.



L'ensemble constitue une tâche



Temps de cycle

$$TC = T_E + T_T + T_S$$

Programmation de l'automate: le Grafcet

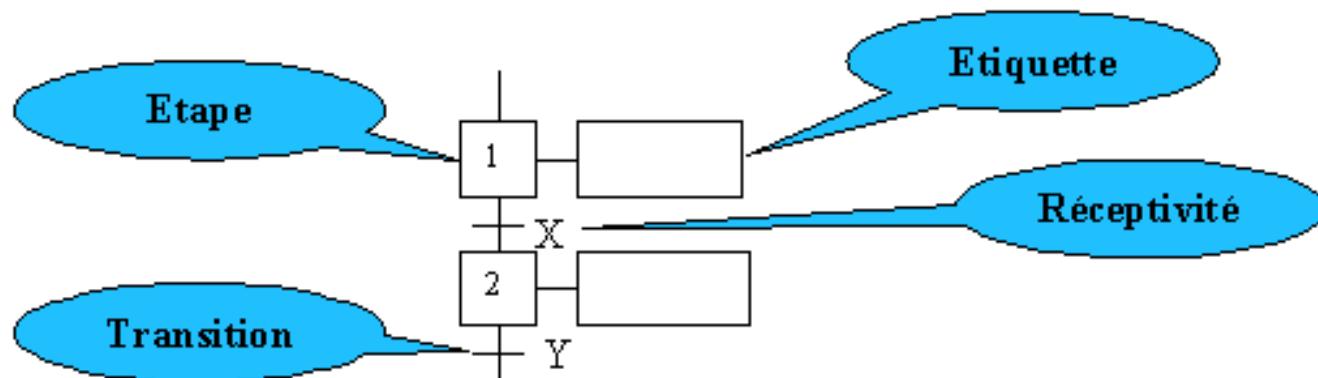
Pourquoi le GRAFCET ?

-
- La création d'une machine automatisée nécessite un dialogue entre le client qui définit le cahier des charges (qui contient les besoins et les conditions de fonctionnement de la machine) et le constructeur qui propose des solutions.
 - Ce dialogue n'est pas toujours facile : le client ne possède peut-être pas la technique lui permettant de définir correctement son problème.
 - D'autre part, le langage courant ne permet pas de lever toutes les ambiguïtés dues au fonctionnement de la machine (surtout si des actions doivent se dérouler simultanément).
 -
 - C'est pourquoi l'**ADEPA** (*Agence pour le Développement de la Productique Appliquée à l'industrie*) a créé le **GRAF CET**.

DÉFINITION

- Le GRAFCET (**G**RAphe **F**onctionnel de **C**ommande des **é**tapes et **T**ransitions) est l'outil de représentation graphique ou modélisation d'un cahier des charges.
- Il a été proposé par l'ADEPA (en 1977 et normalisé en 1982 par la NF C03-190).

Le GRAFCET est une représentation alternée d'**étapes** et de **transitions**. Une seule transition doit séparer deux étapes.



Une **étape** correspond à une situation dans laquelle les variables de sorties conservent leur état.

Une **transition** indique la possibilité d'évolution entre deux étapes successives. A chaque transition est associée une condition logique appelée **réceptivité**.

Les avantages du GRAFCET ?

Les avantages du GRAFCET ?

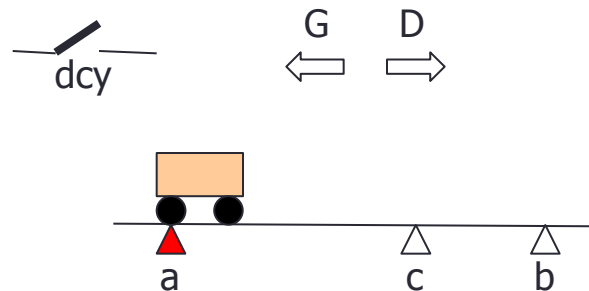
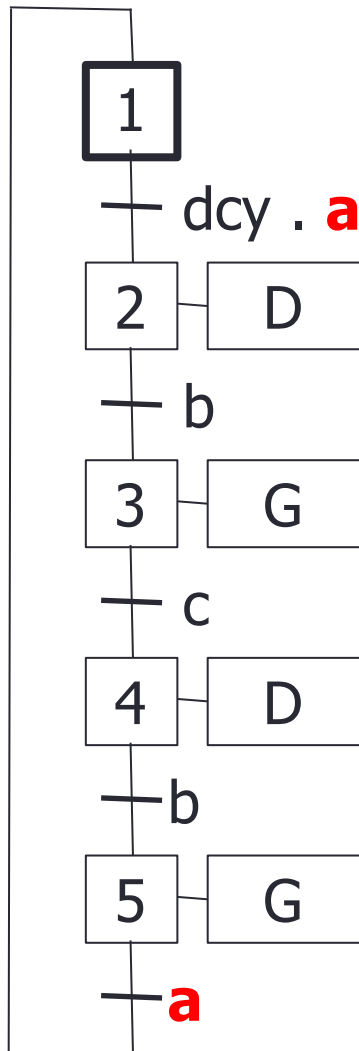
il est indépendant de la *matérialisation technologique*

il traduit de façon cohérente le *cahier des charges*

il est bien adapté à la complexité des *systèmes automatisés*

il est bien adapté à la **spécification, conception et réalisation**

Exemple d'application



Cahier des charges:

Après l'ordre de départ cycle « dcy », le chariot part jusque b, revient en c, repart en b puis rentre en a

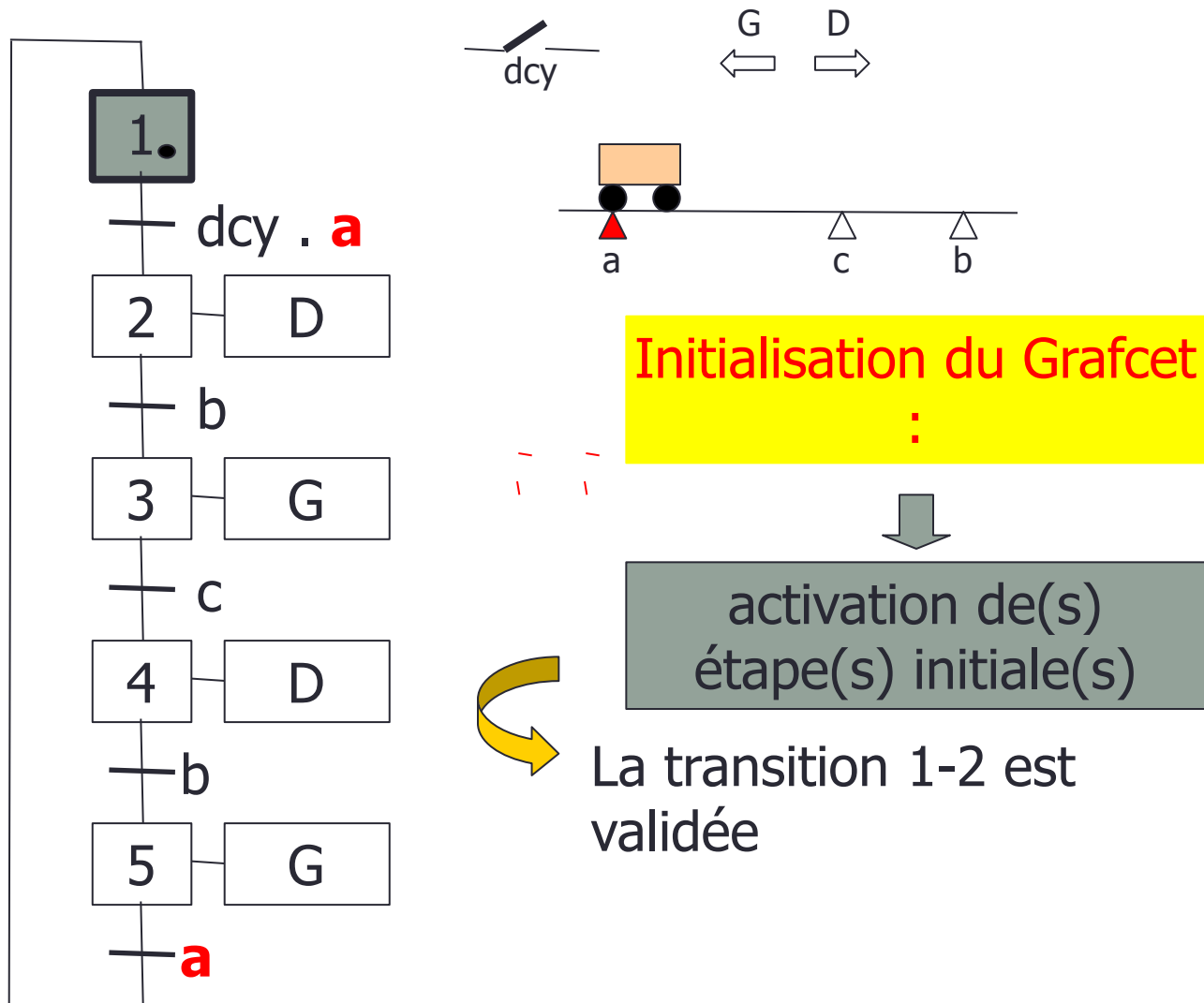
Capteurs:

- a : chariot à gauche
- b : chariot à droite

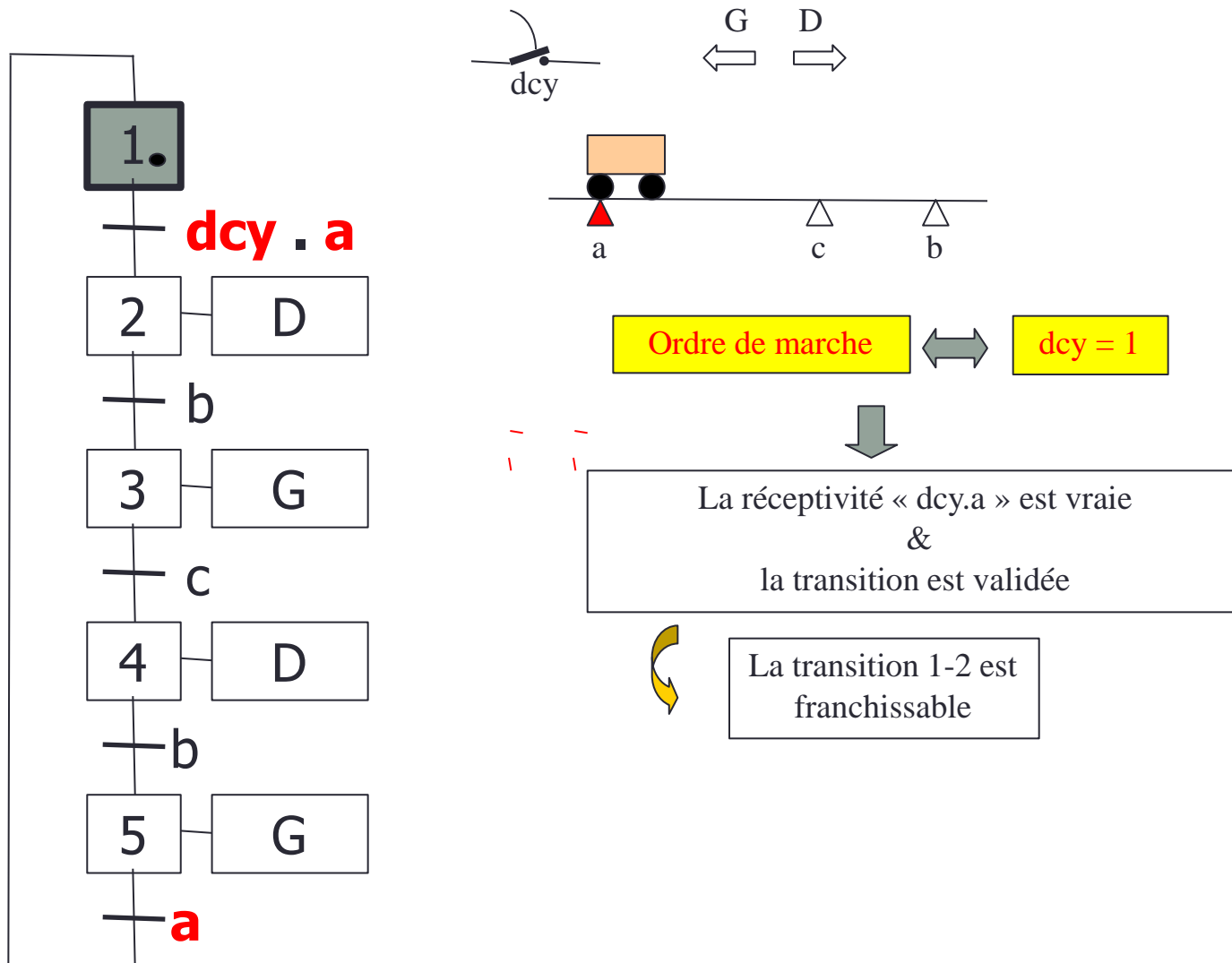
Actionneurs:

- D : aller à droite
- G : aller à gauche

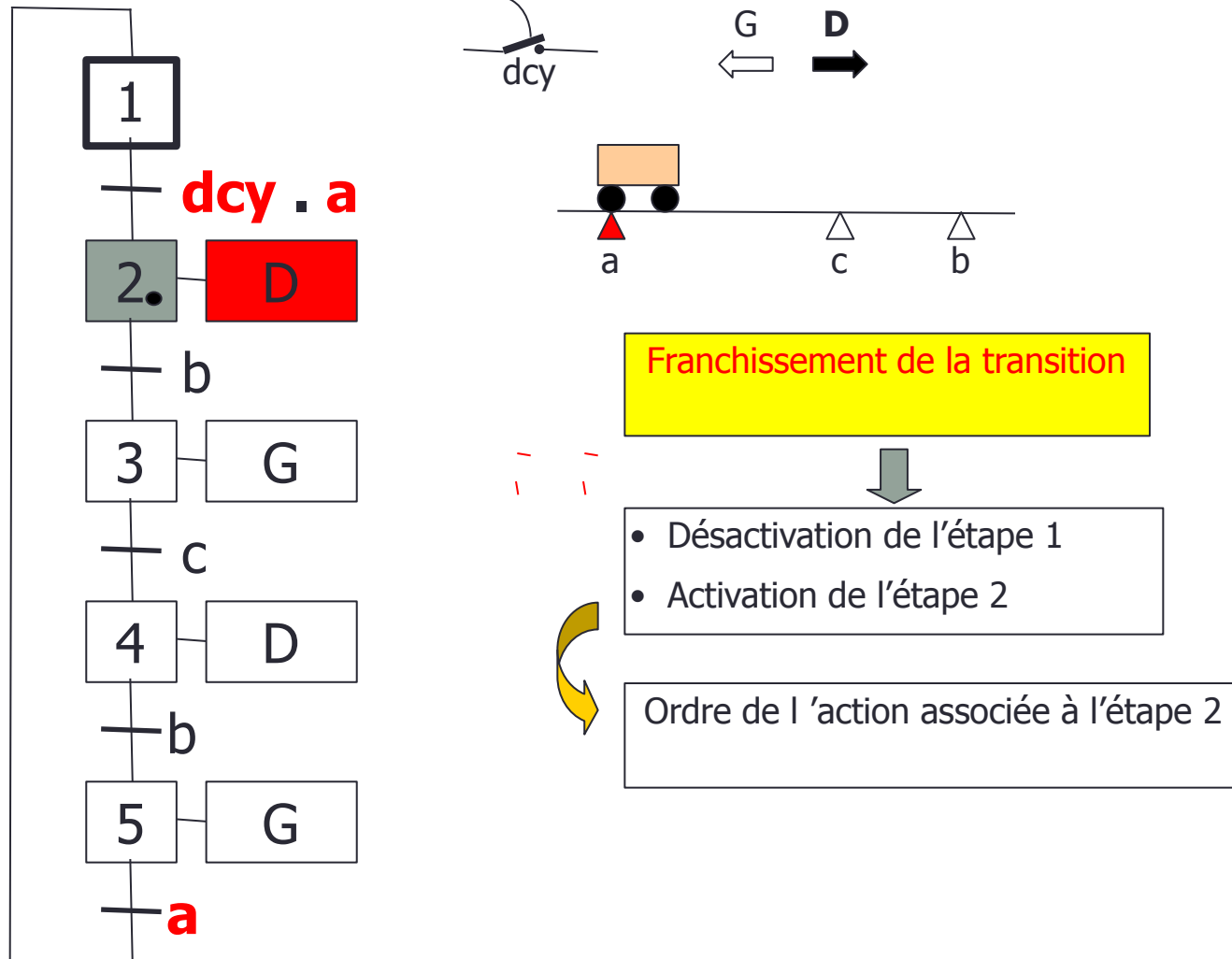
Exemple d'application



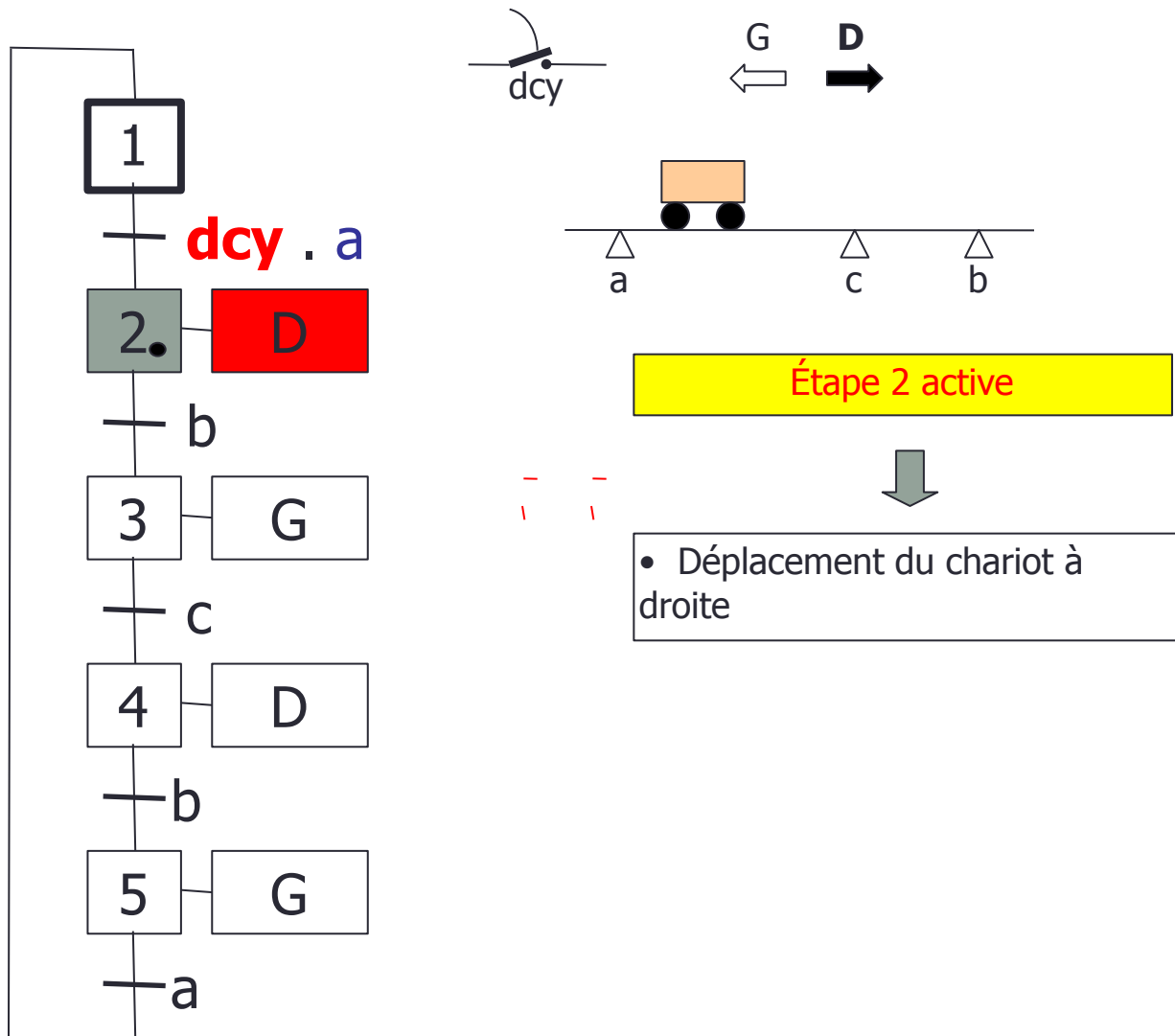
Exemple d'application



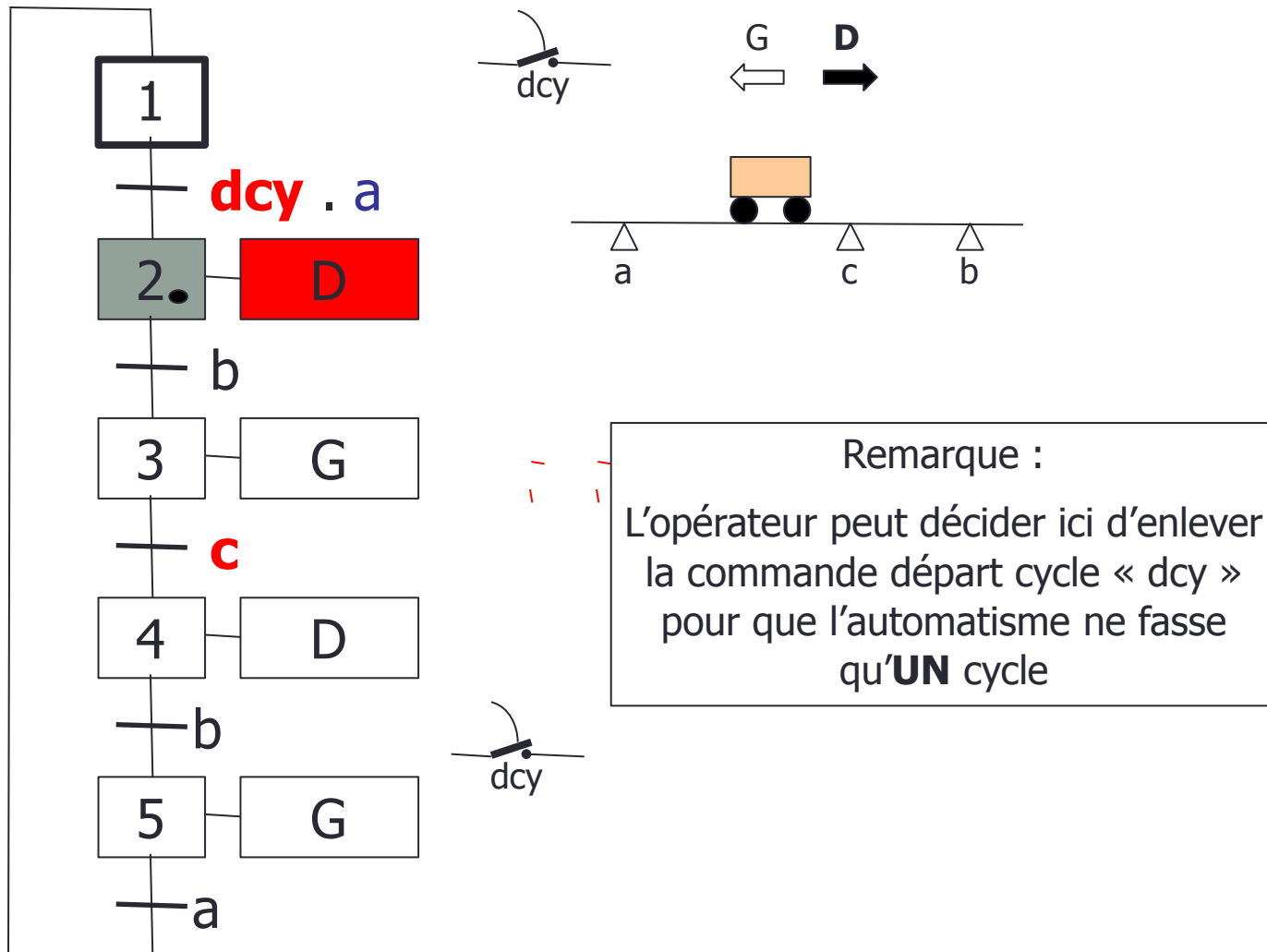
Exemple d'application



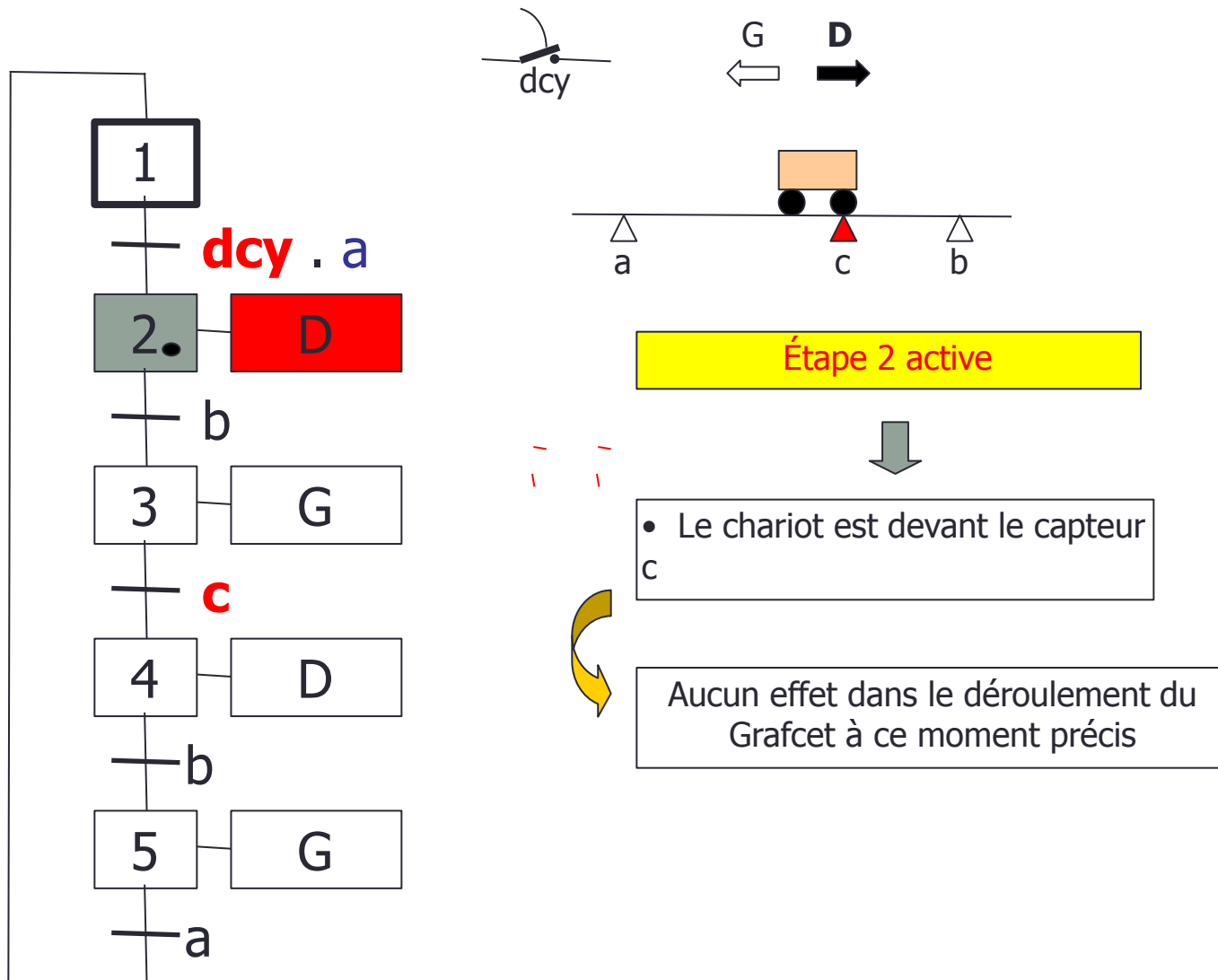
Exemple d'application



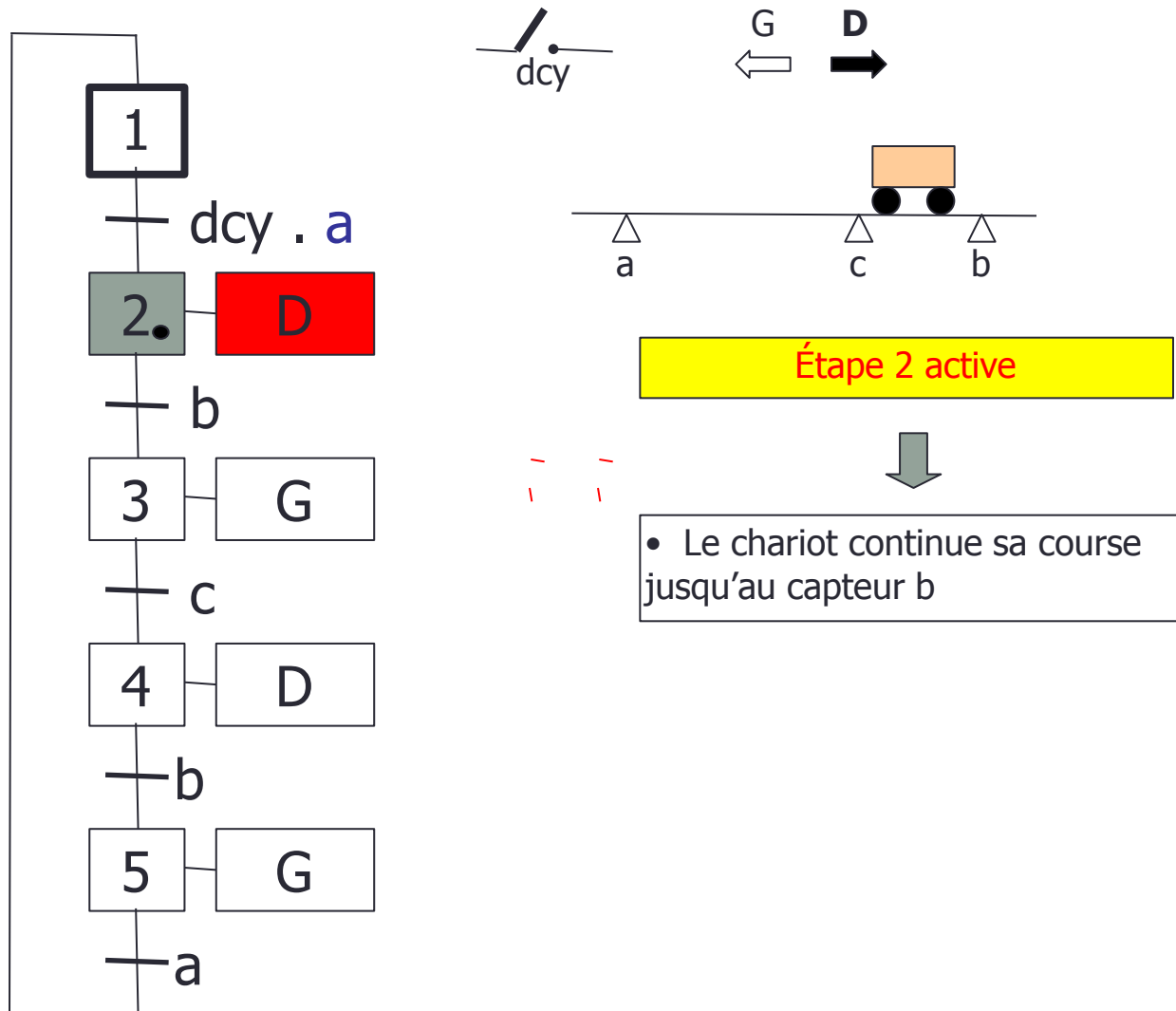
Exemple d'application



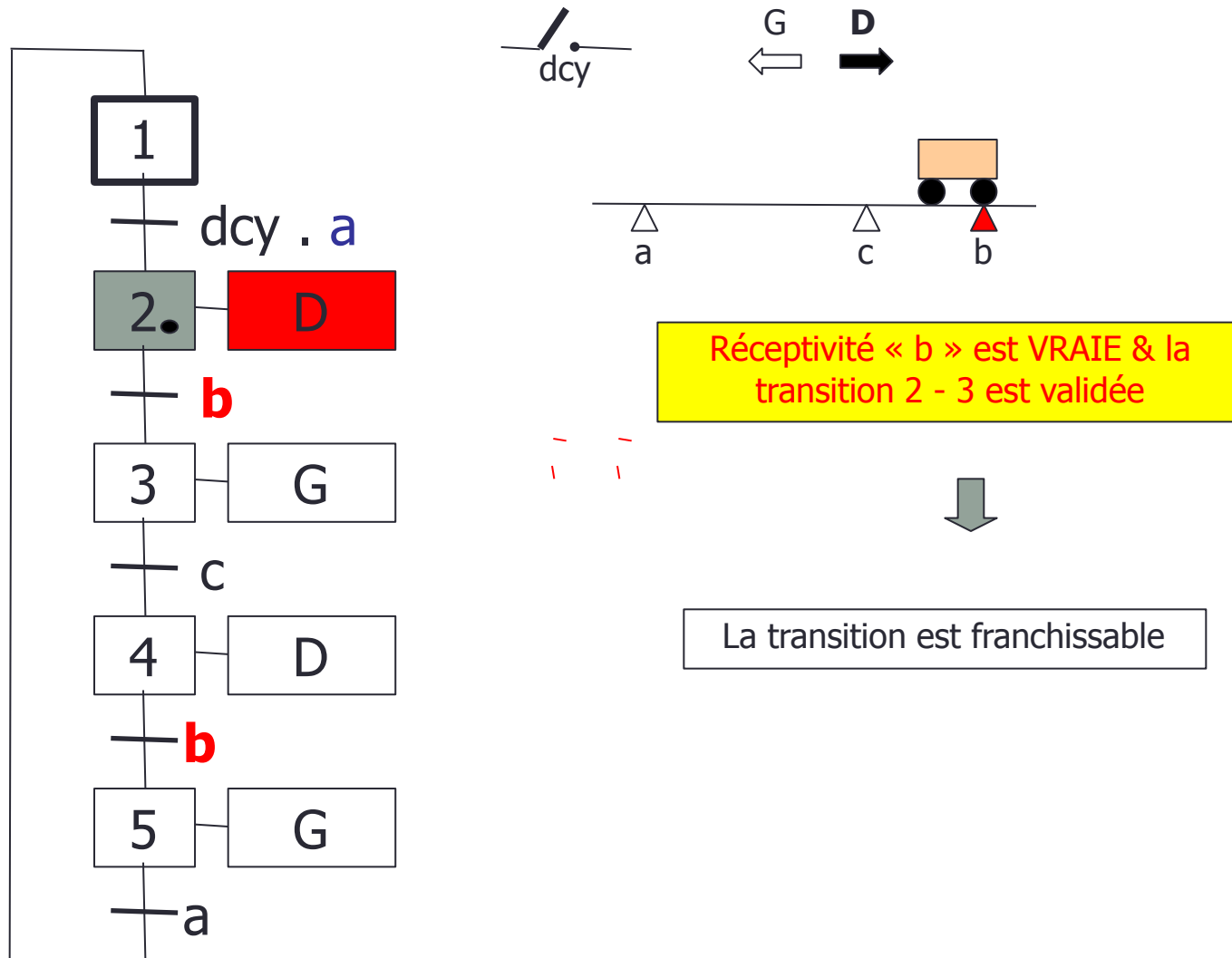
Exemple d'application



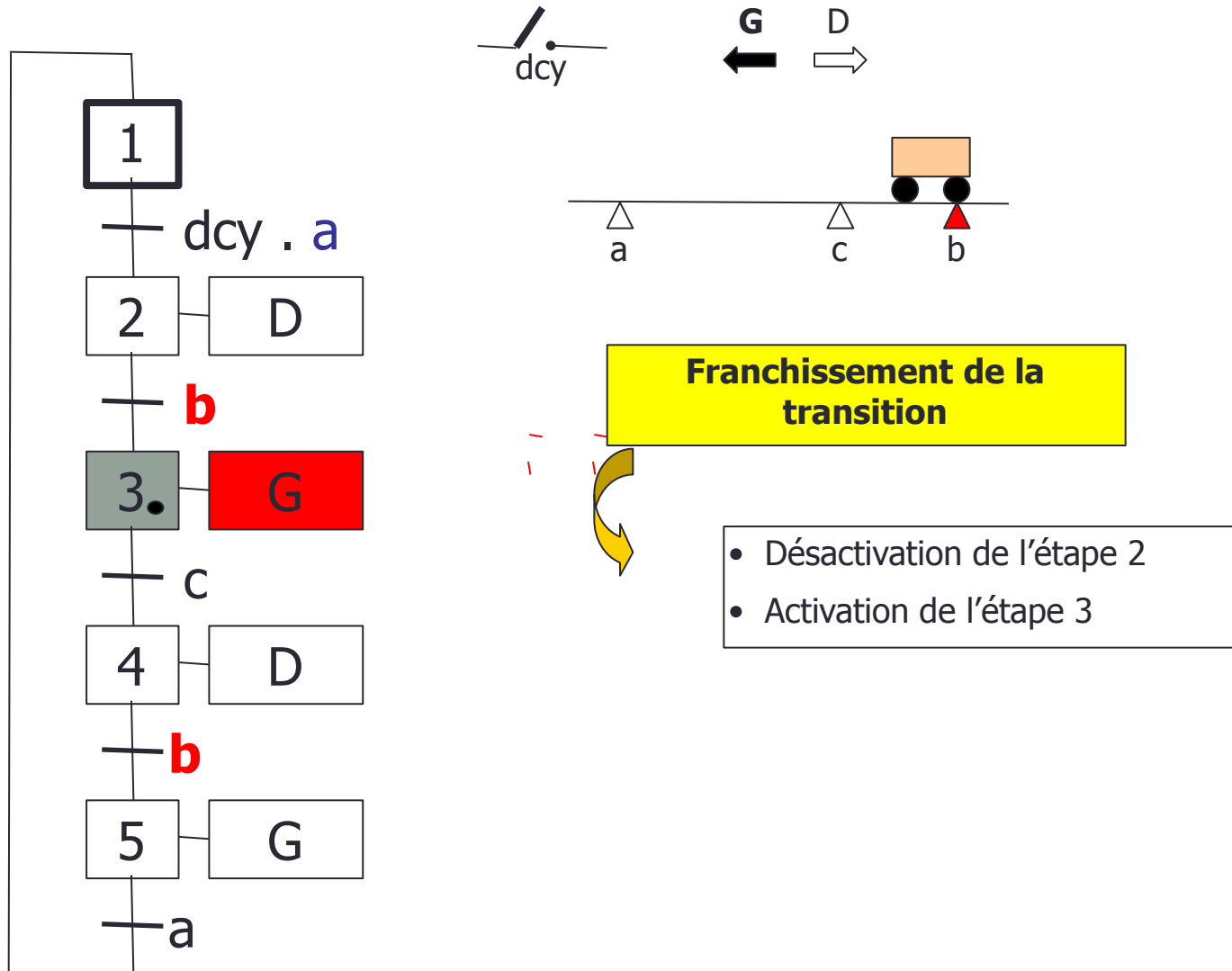
Exemple d'application



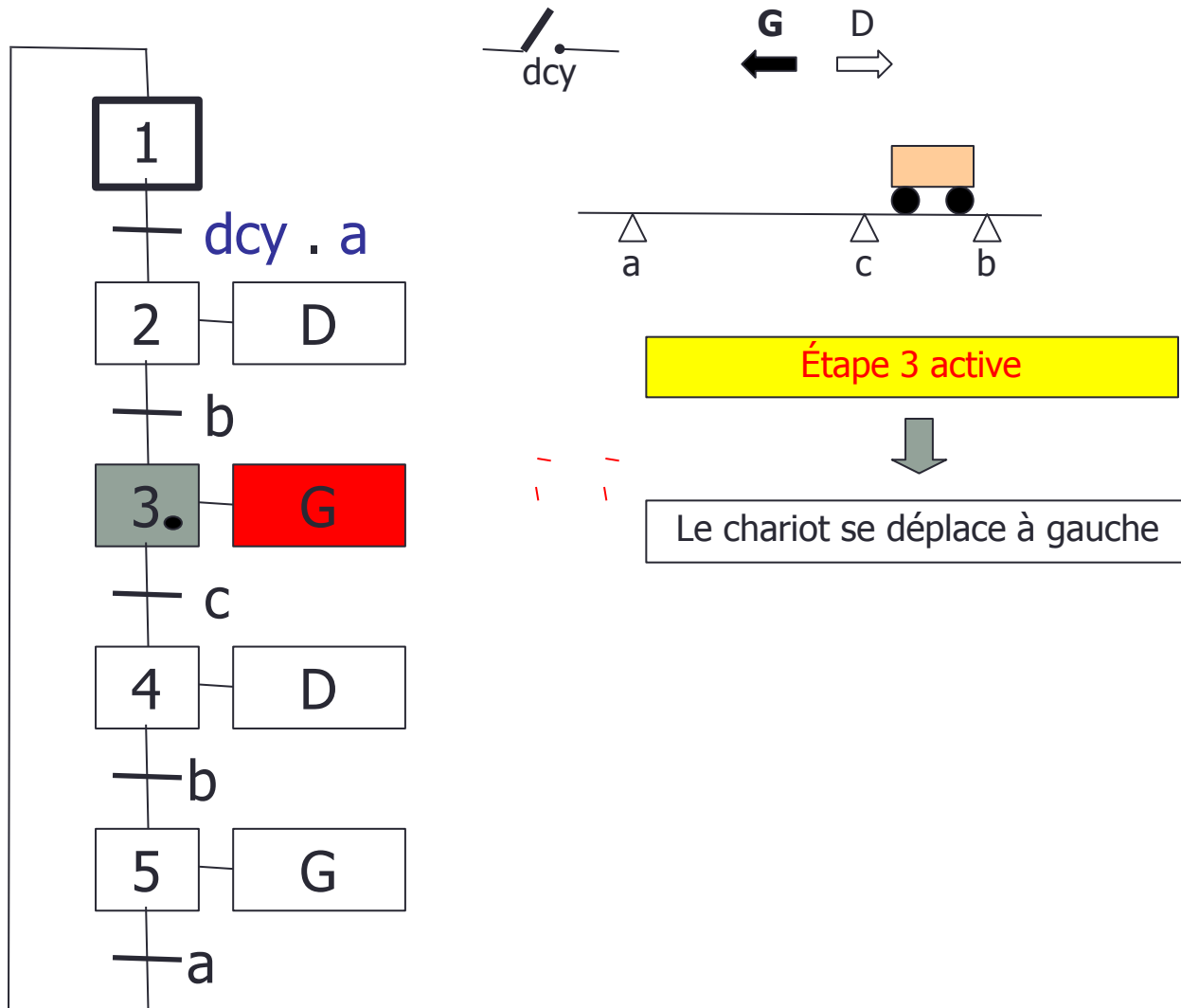
Exemple d'application



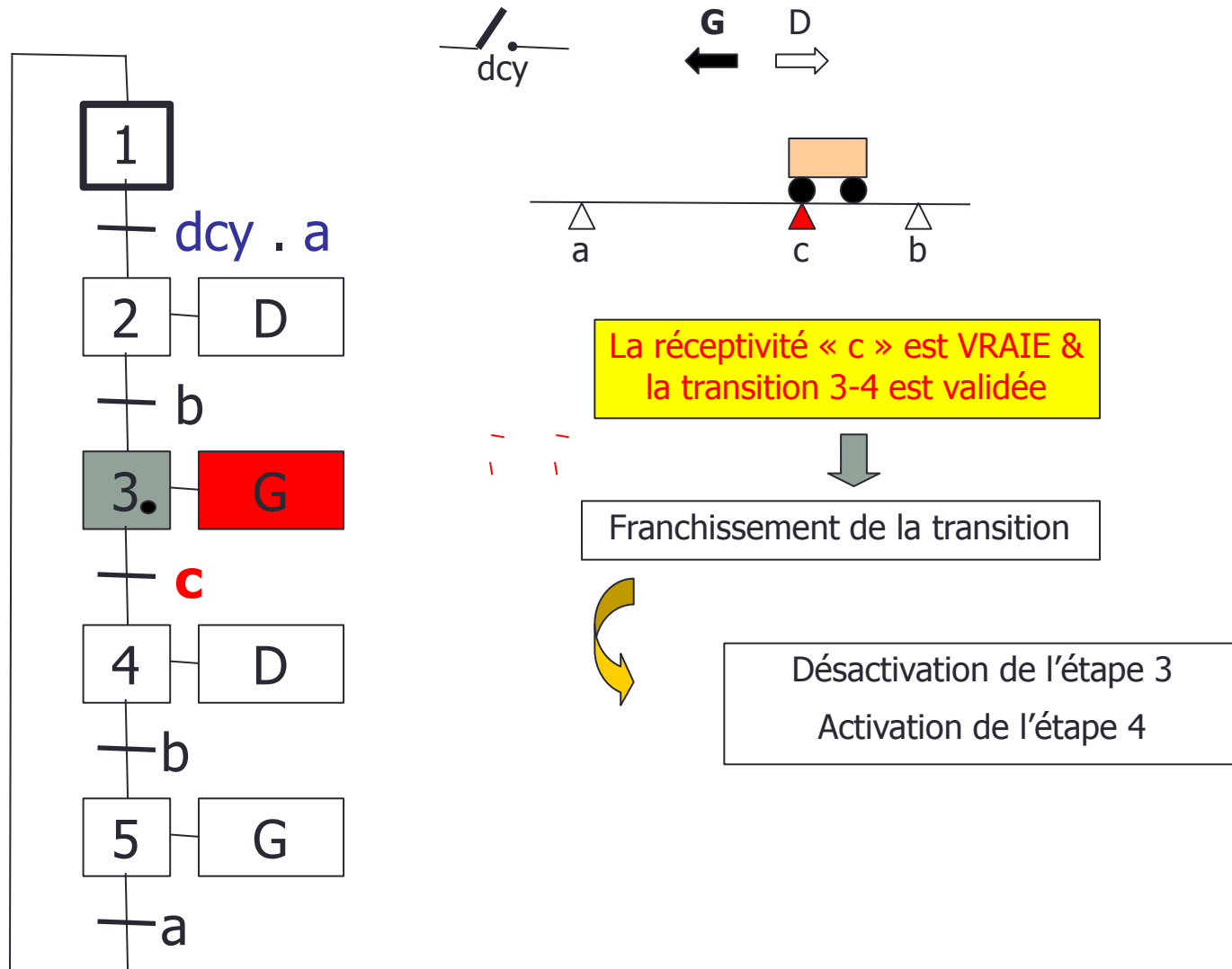
Exemple d'application



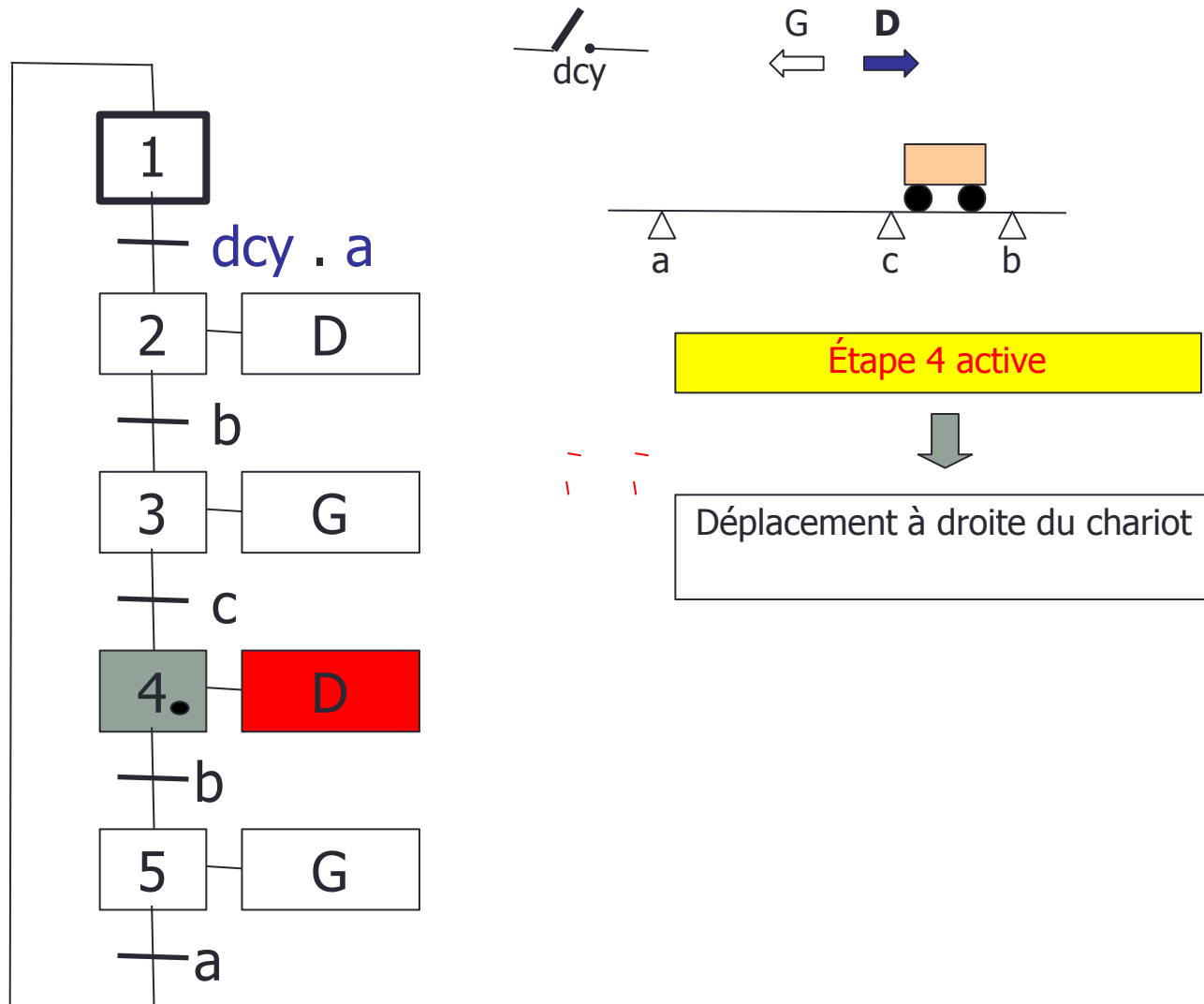
Exemple d'application



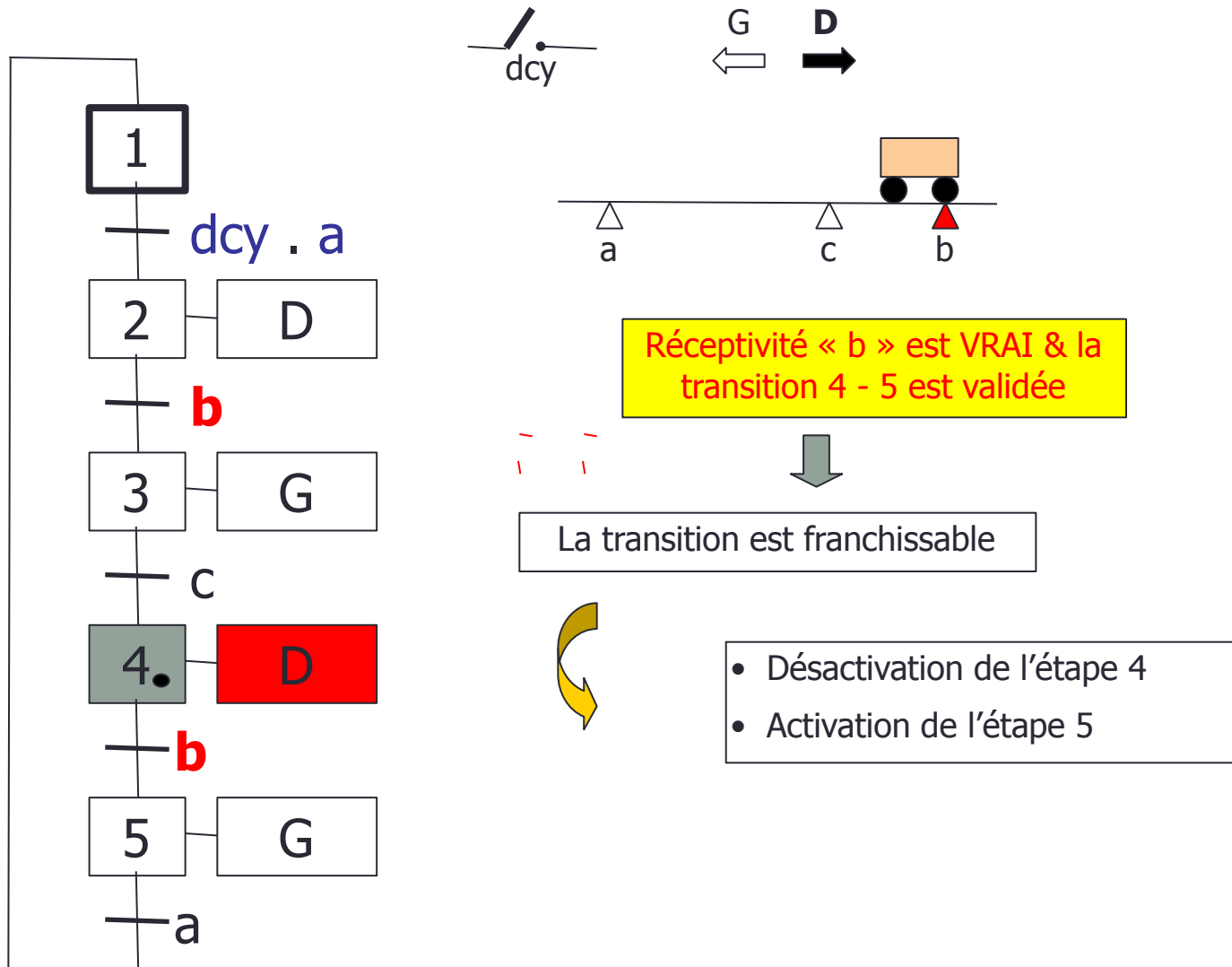
Exemple d'application



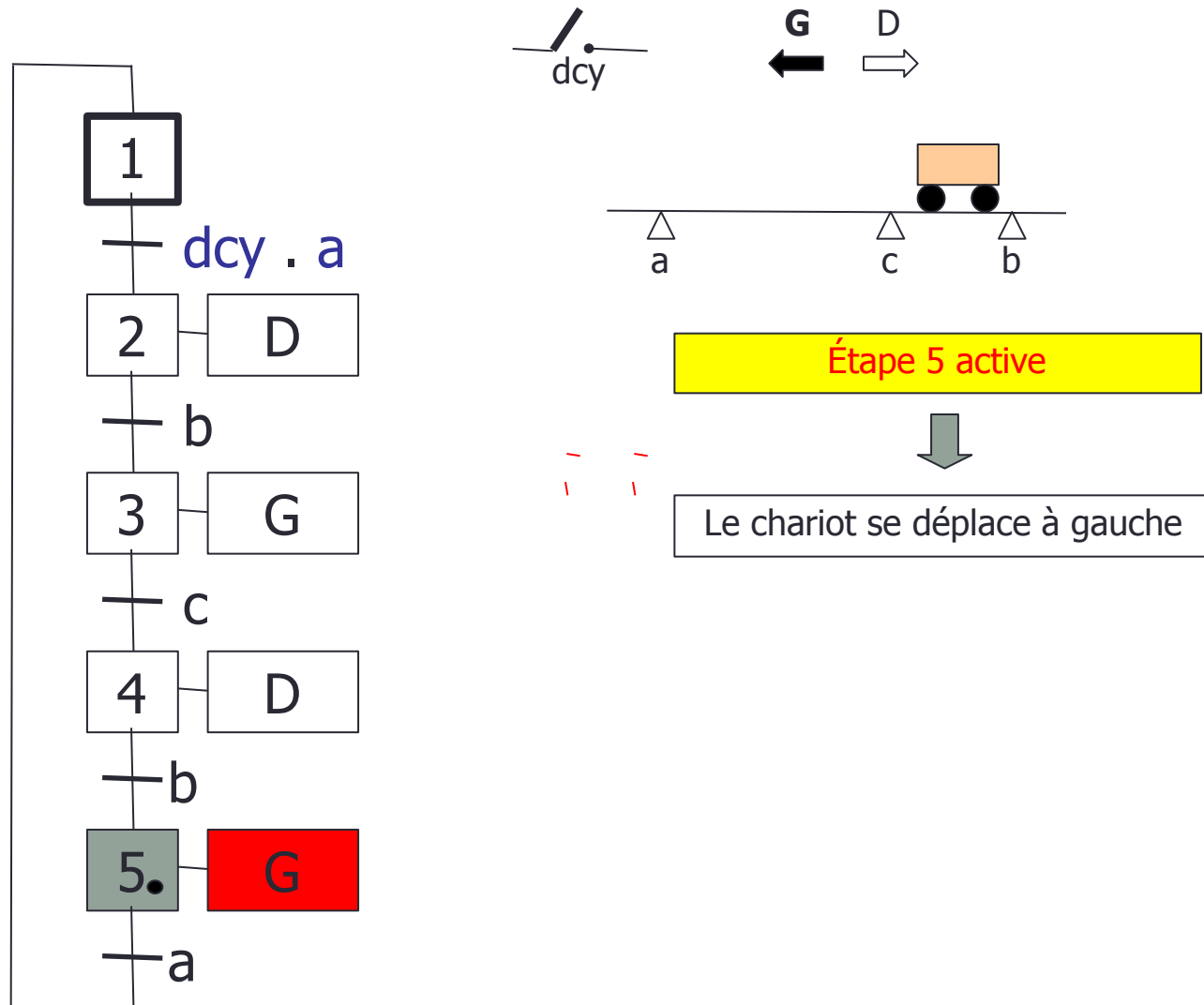
Exemple d'application



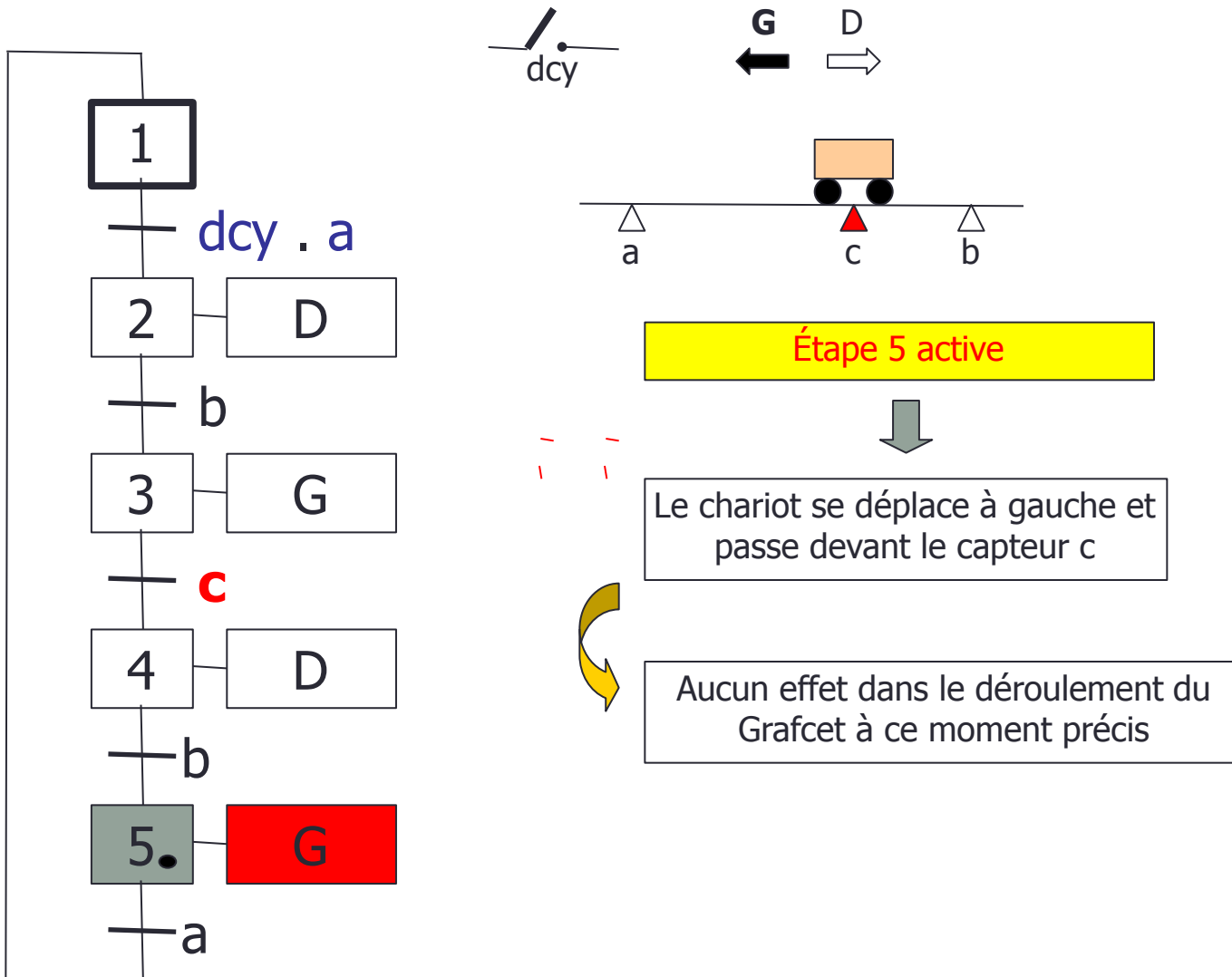
Exemple d'application



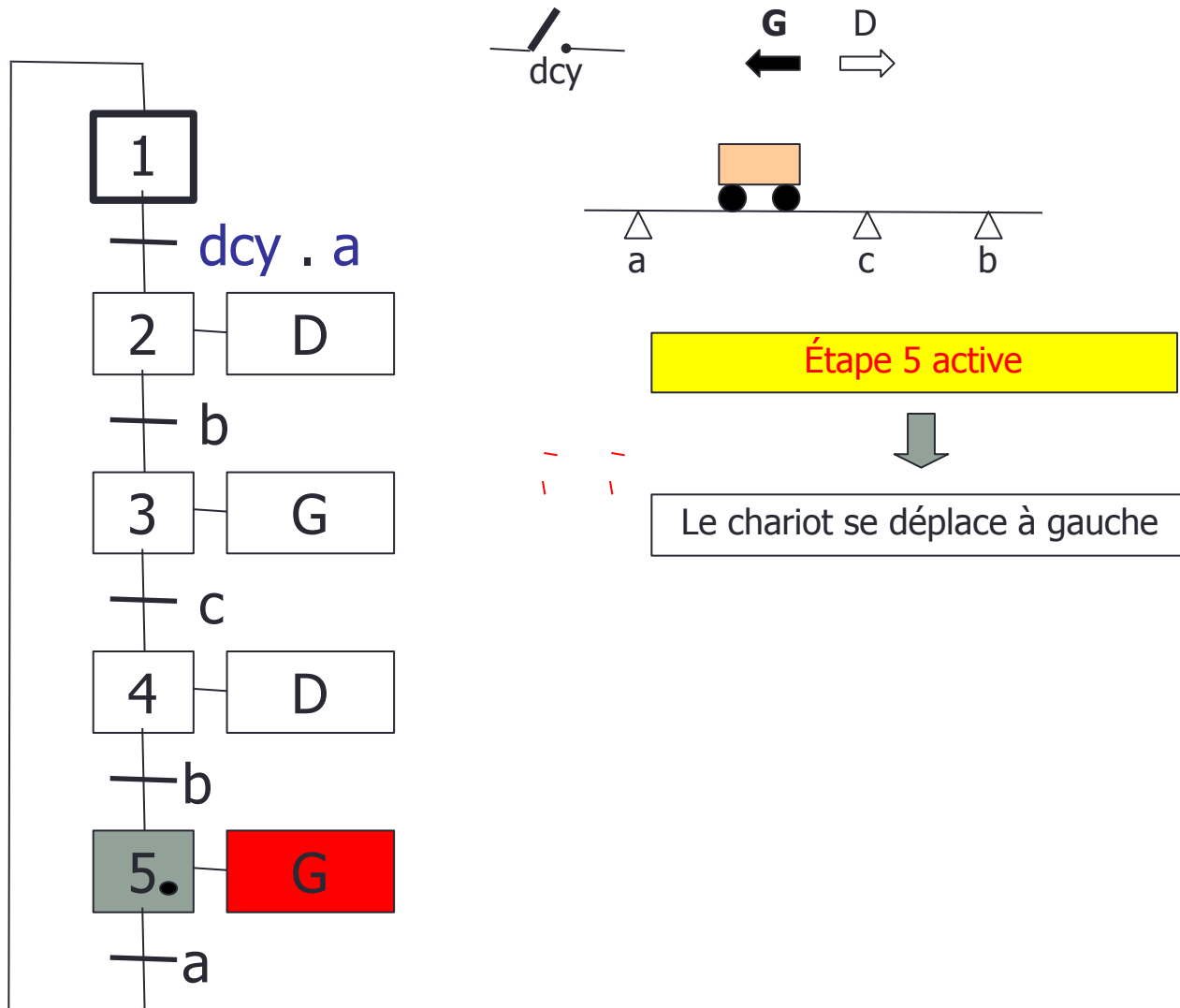
Exemple d'application



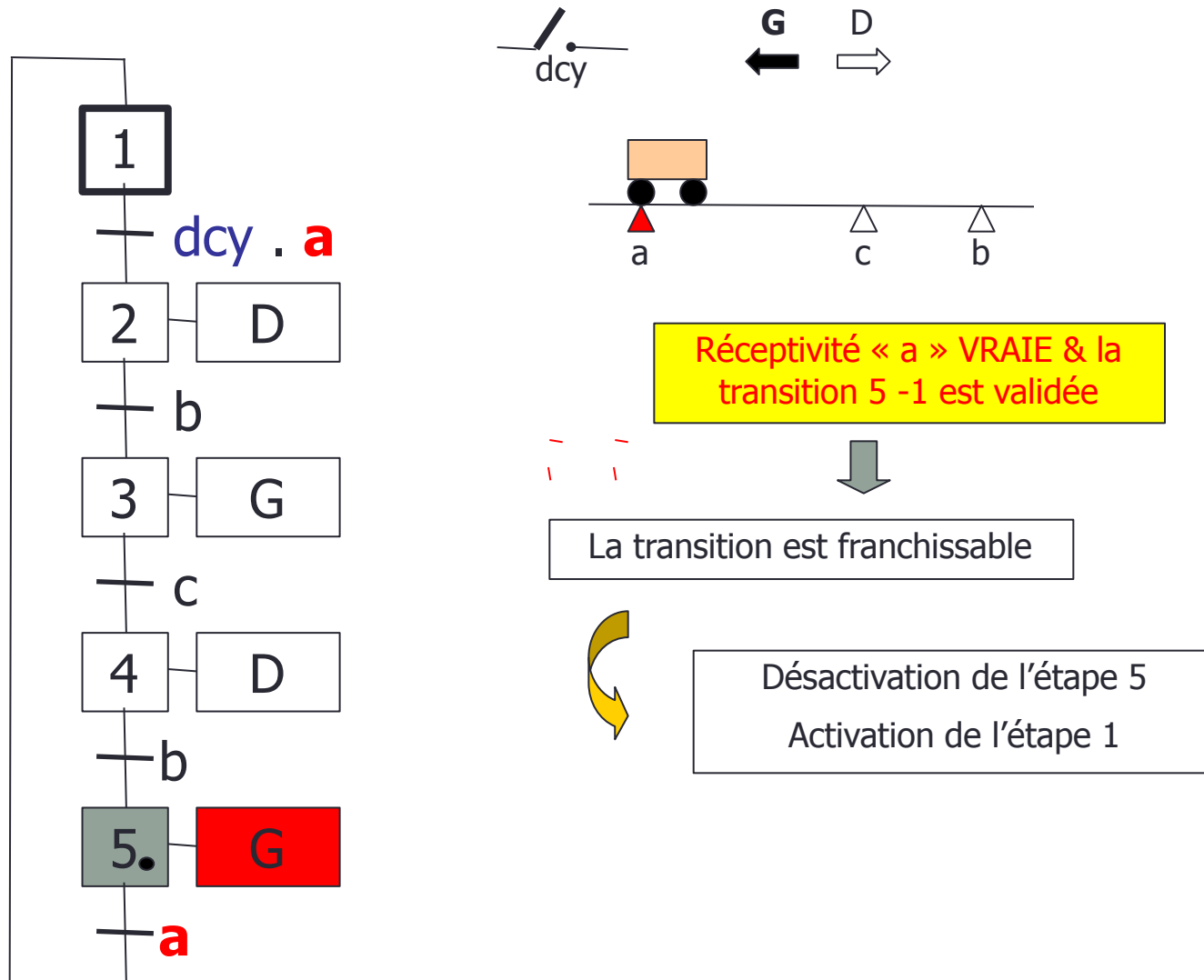
Exemple d'application



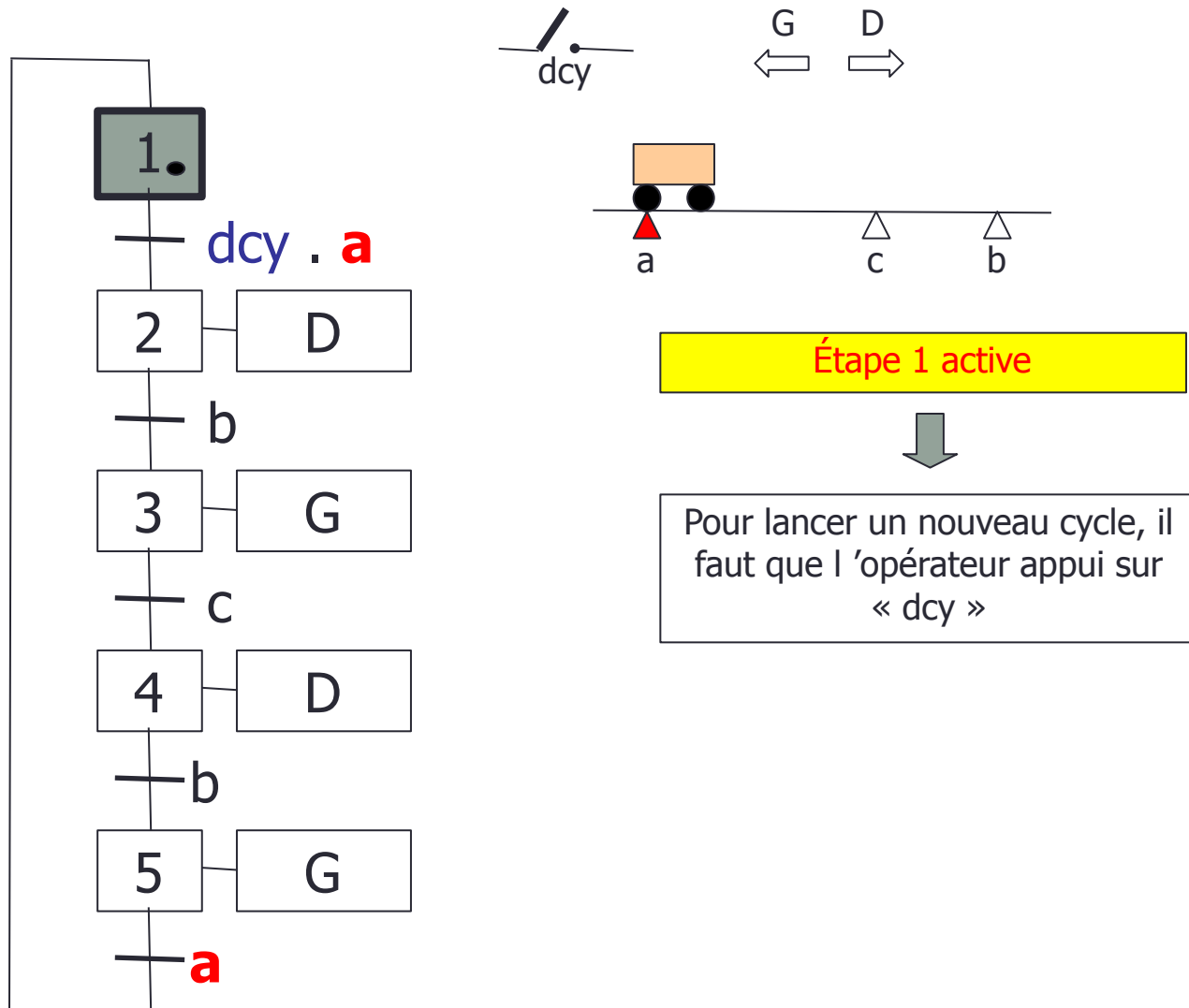
Exemple d'application



Exemple d'application

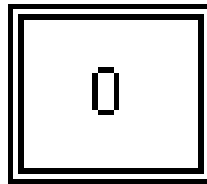


Exemple d'application



RÈGLES DE SYNTAXE

Règle N°1 : situation initiale



Cette représentation indique que l'étape est initialement activée (à la mise sous tension de la partie commande).

La situation initiale, choisie par le concepteur, est la **situation à l'instant initial ou au repos.**

Règle N°2 :

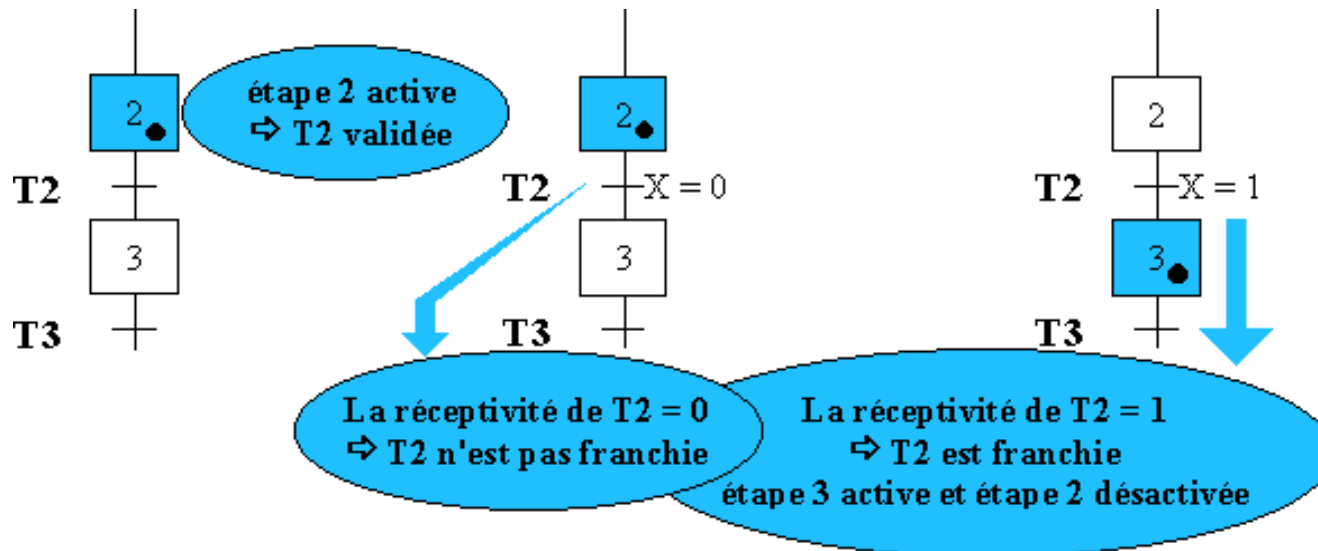
franchissement d'une transition

Une transition est **franchie** lorsque l'étape associée est **active** et la **réceptivité** associée à cette transition est **vraie**.

Règle N°3 :

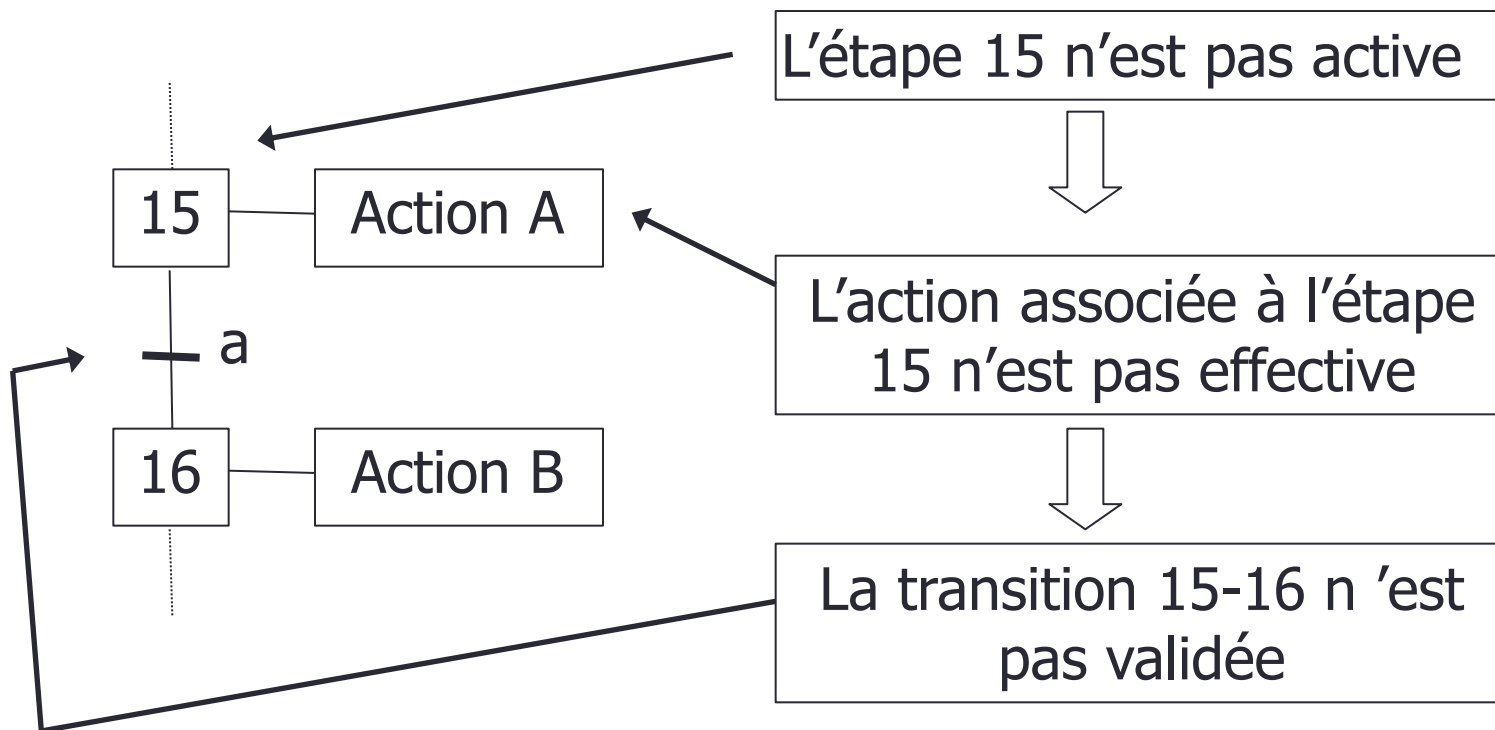
Evolution des étapes actives

- Le franchissement d'une transition provoque simultanément :
 - la **désactivation** de toutes les étapes immédiatement précédentes reliées à cette transition,
 - l'**activation** de toutes les étapes immédiatement suivantes reliées à cette transition.

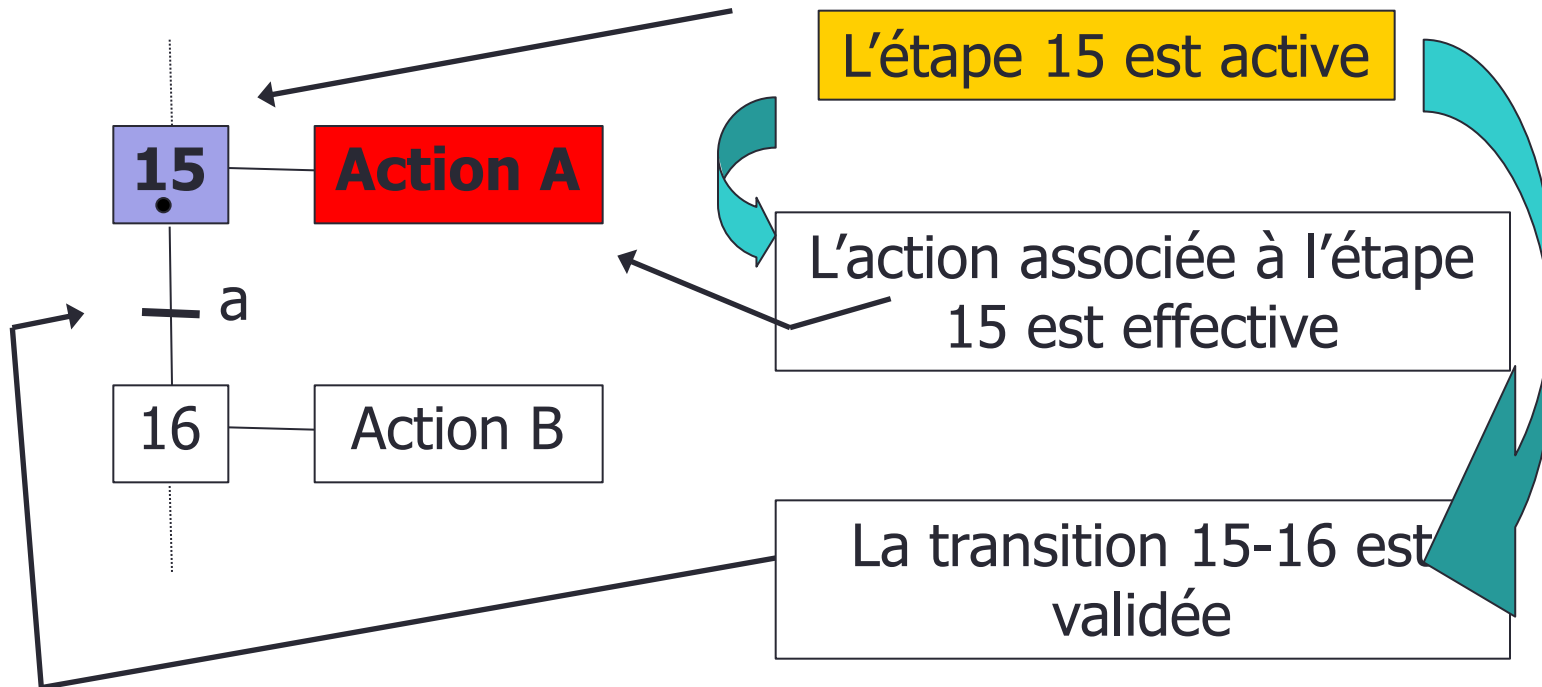


Principe d'évolution

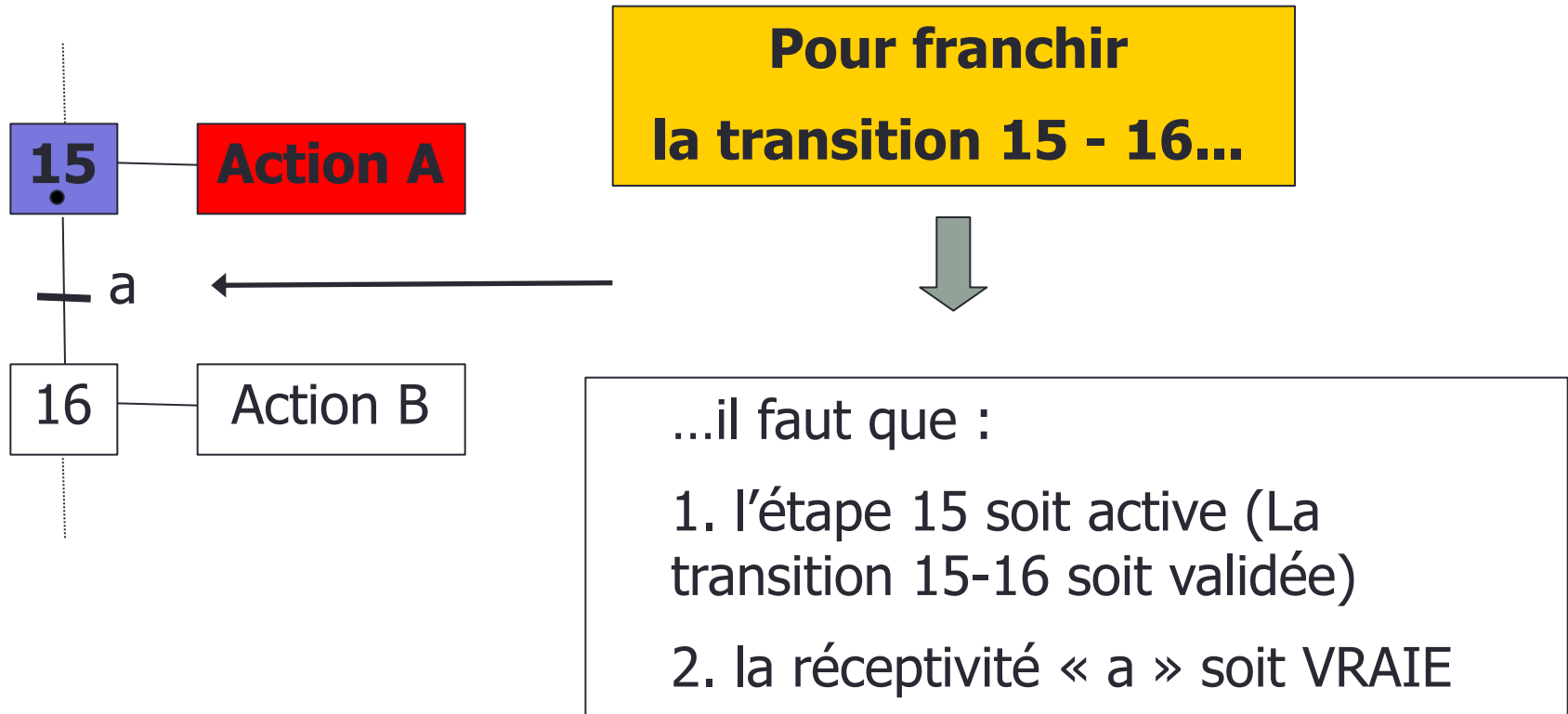
illustration : franchissement d'une transition



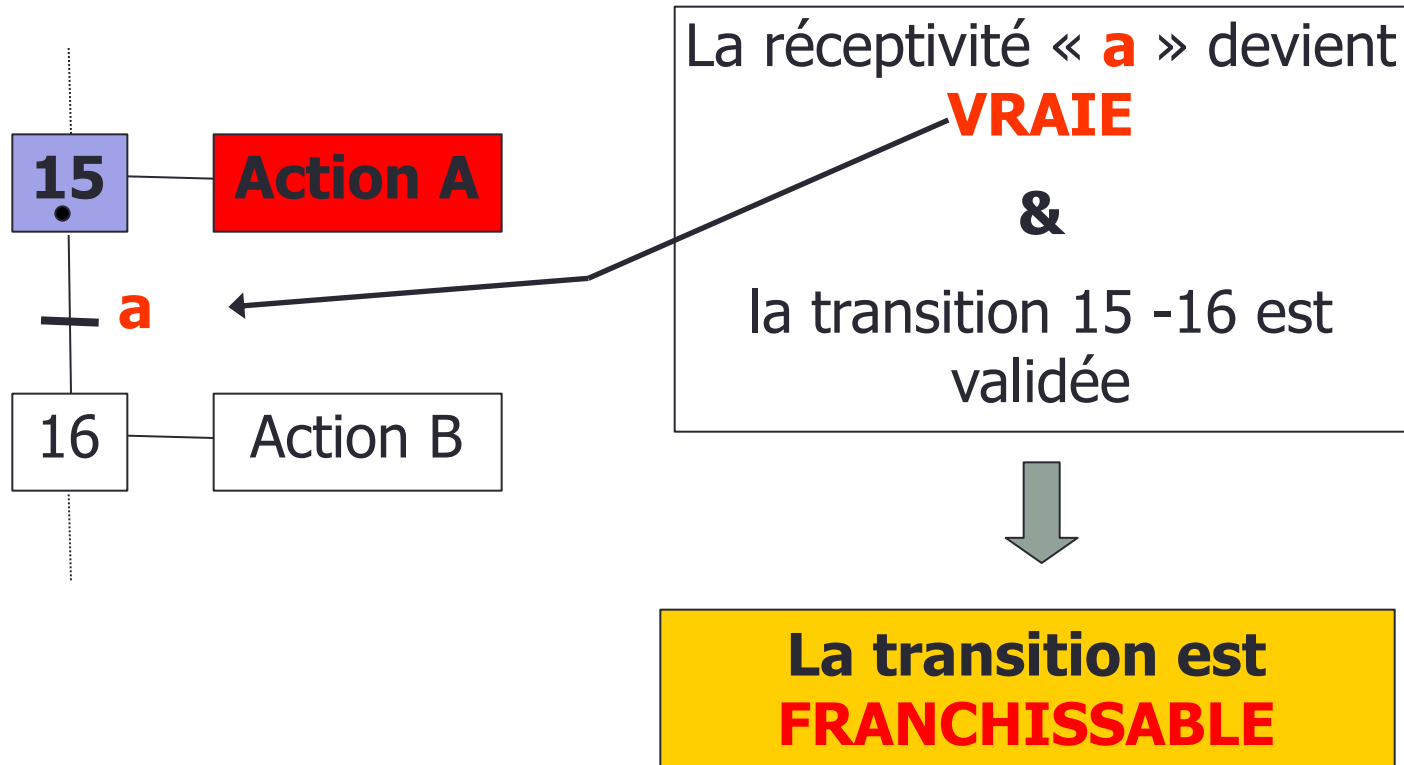
Principe d'évolution



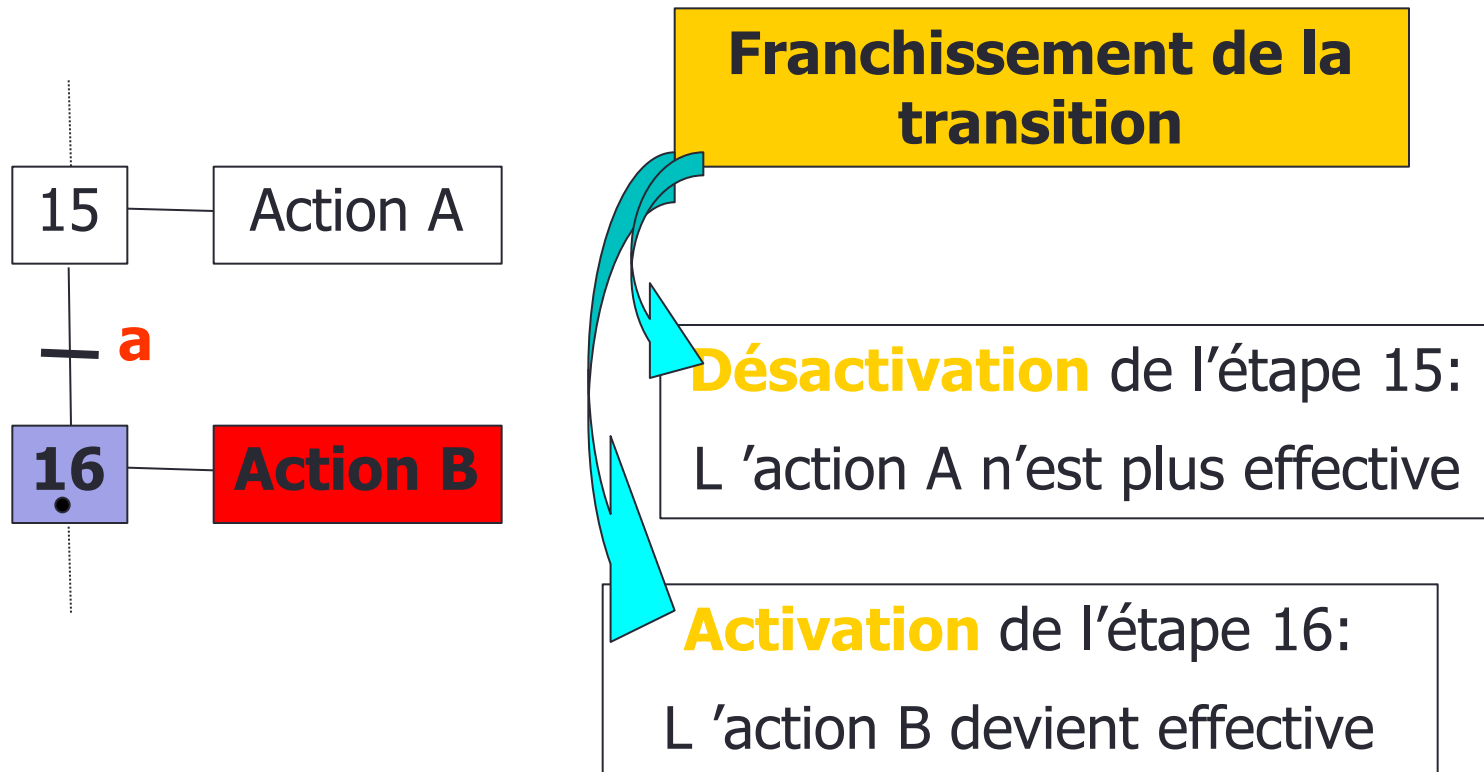
Principe d'évolution



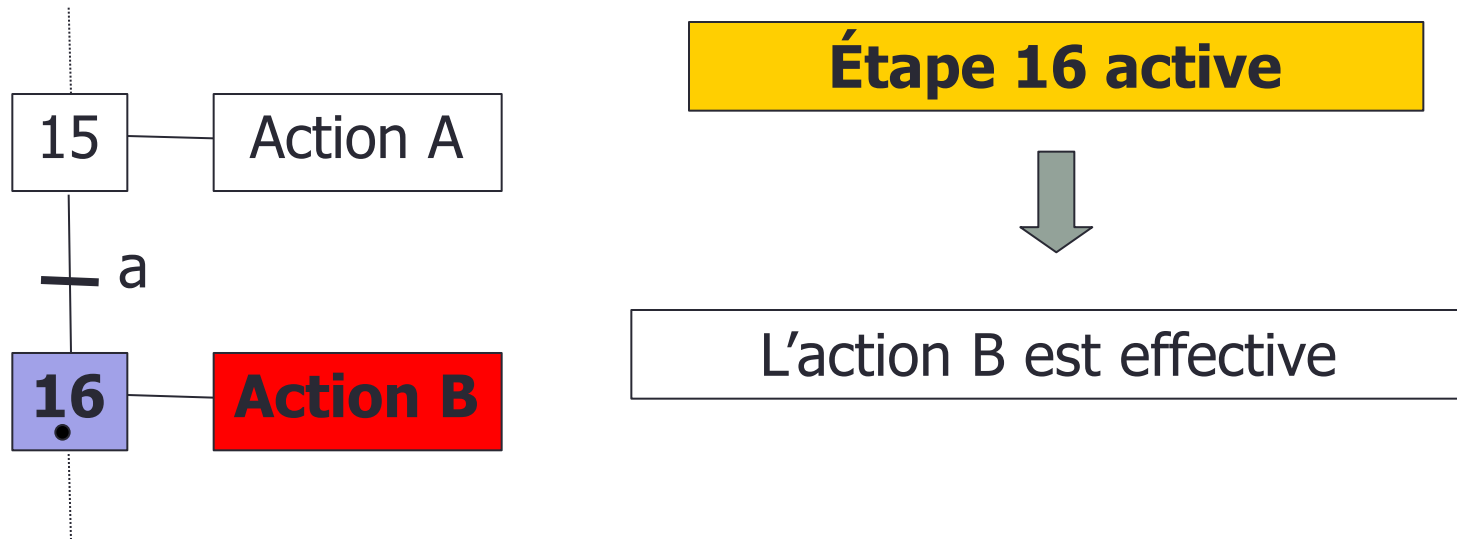
Principe d'évolution



Principe d'évolution



Principe d'évolution

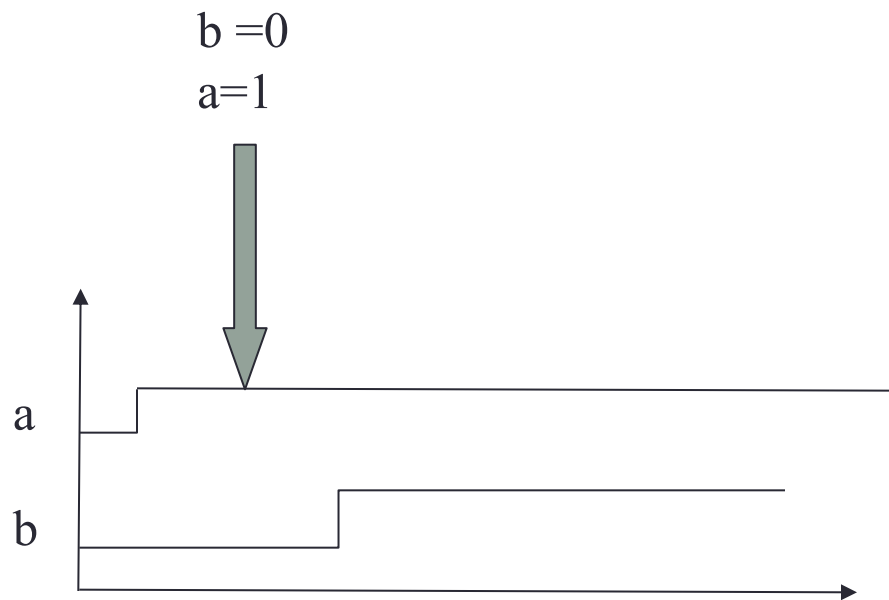
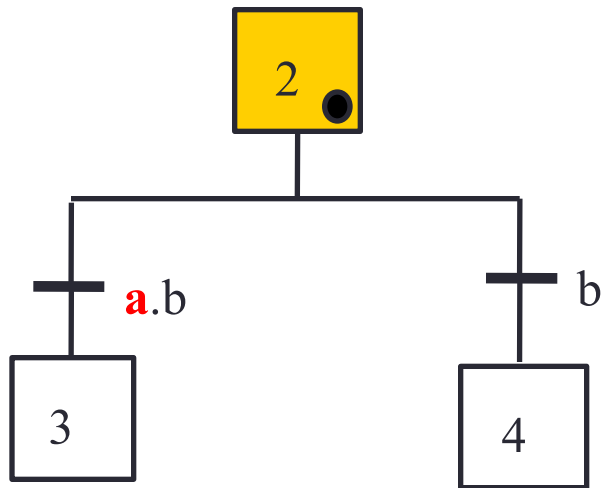


Remarque : la réceptivité « a », quelle soit VRAIE ou FAUSSE à ce moment n'a plus d'effet sur le déroulement du Grafcet

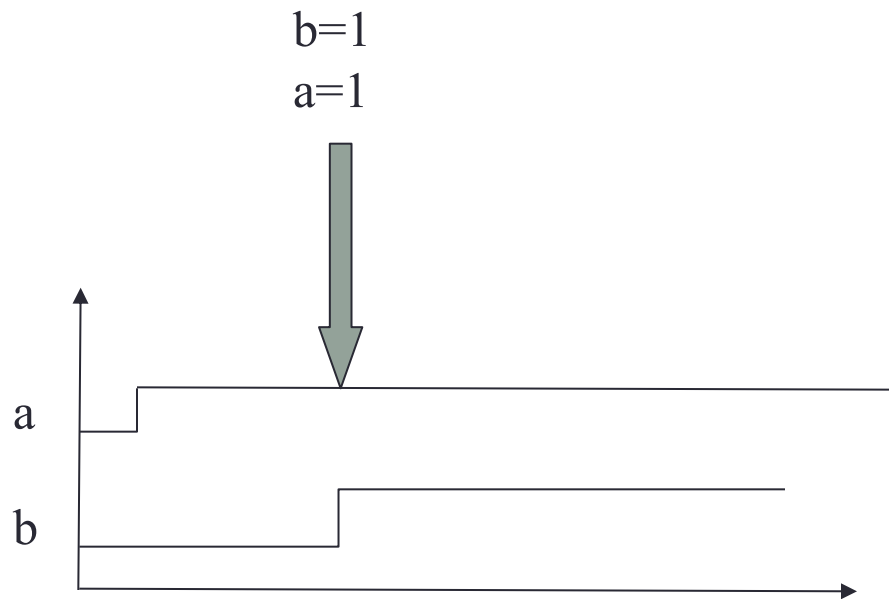
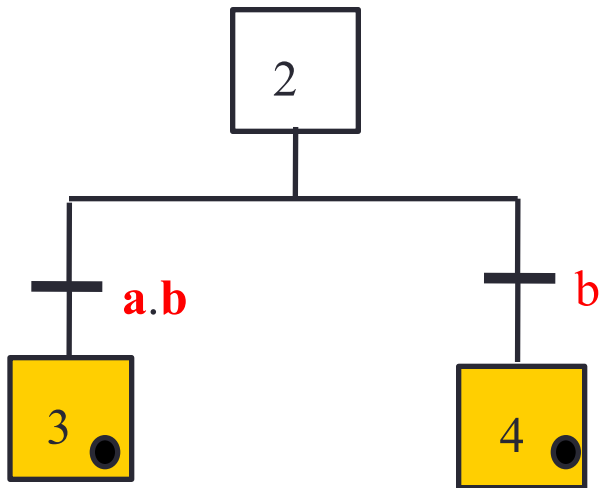
3 Règles de franchissement

- Toute transition franchissable est immédiatement franchie
- Plusieurs transitions simultanément franchissables sont simultanément franchies
- Lorsqu'une étape est simultanément activée et désactivée, elle reste active

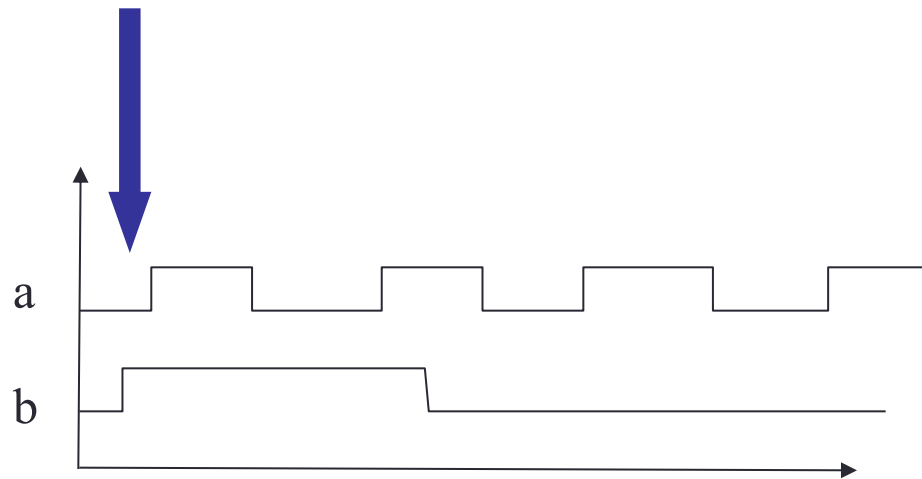
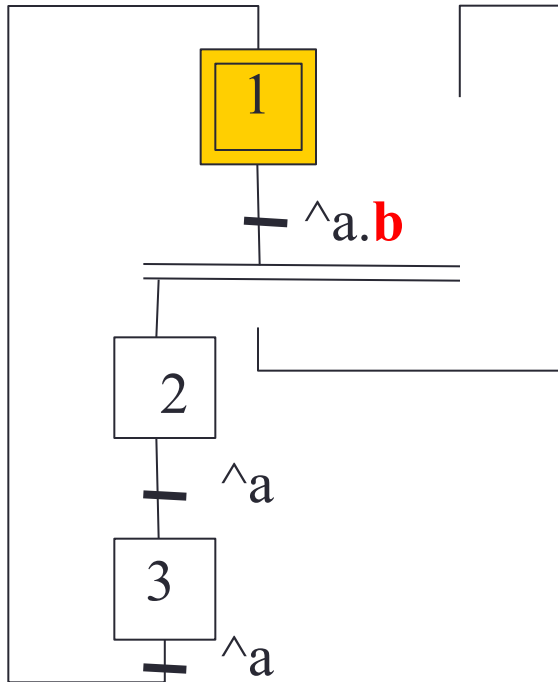
Plusieurs transitions simultanément franchissables sont simultanément franchies



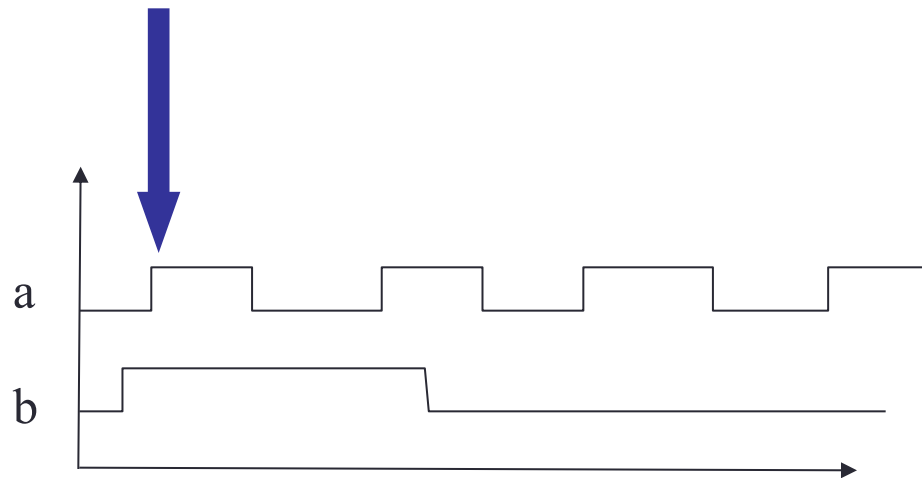
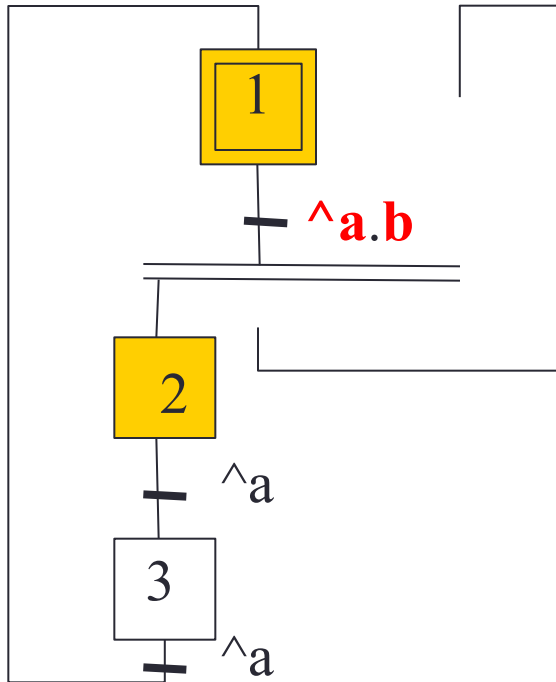
Plusieurs transitions simultanément franchissables sont simultanément franchies



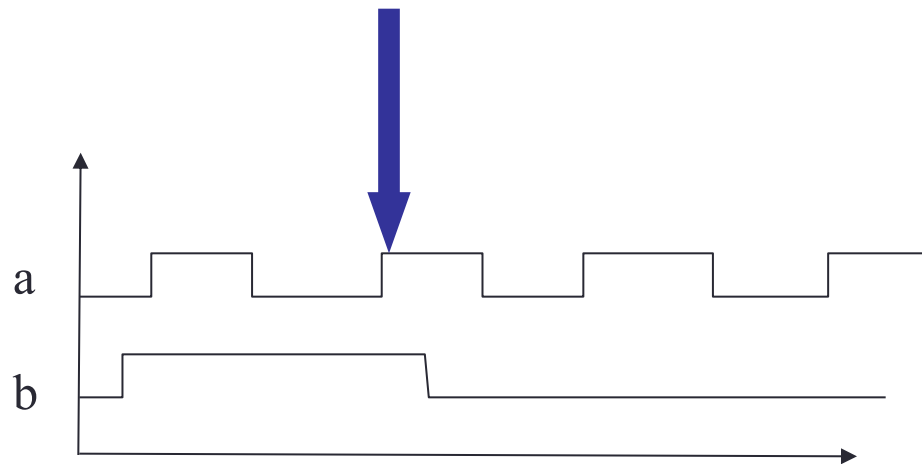
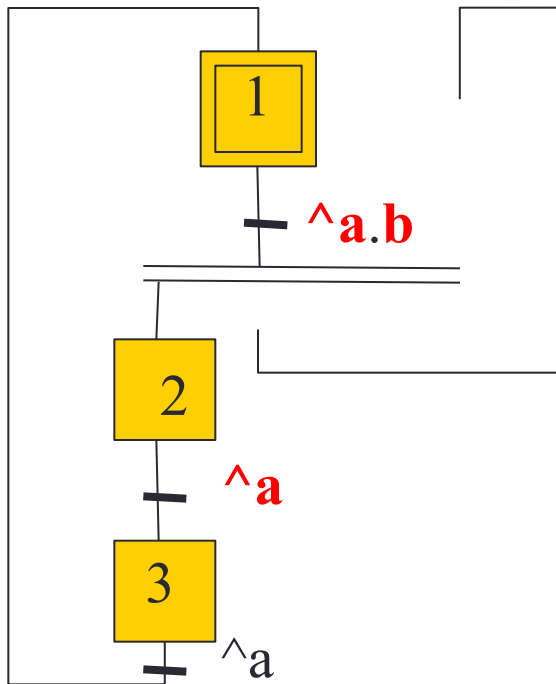
Etape simultanément activée et désactivée



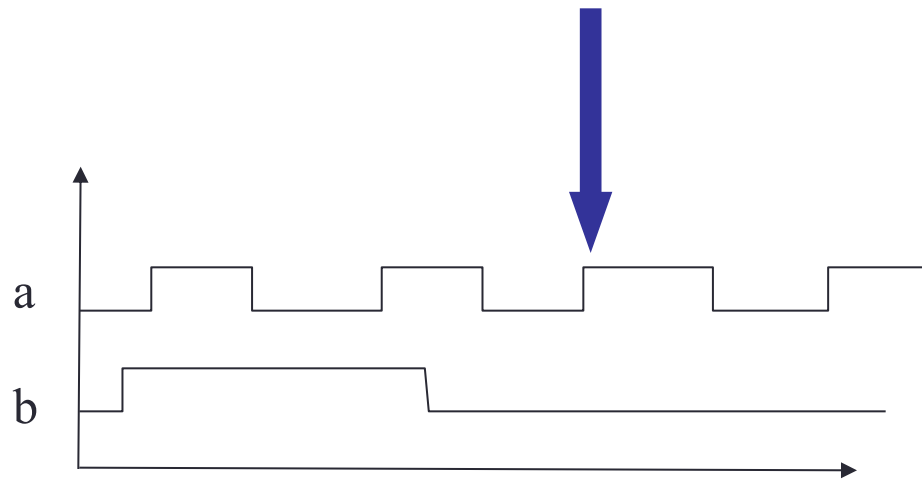
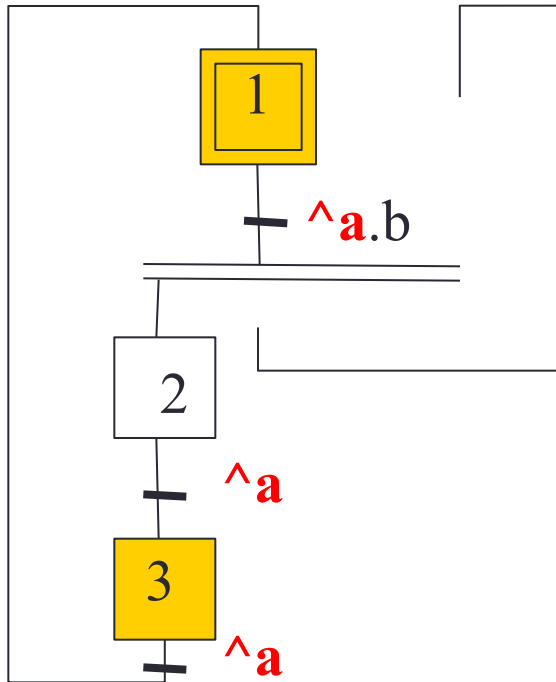
Etape simultanément activée et désactivée



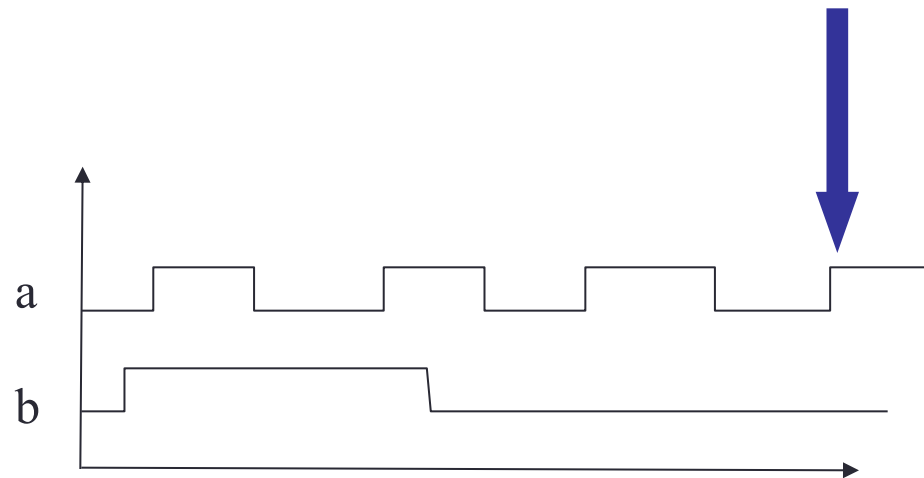
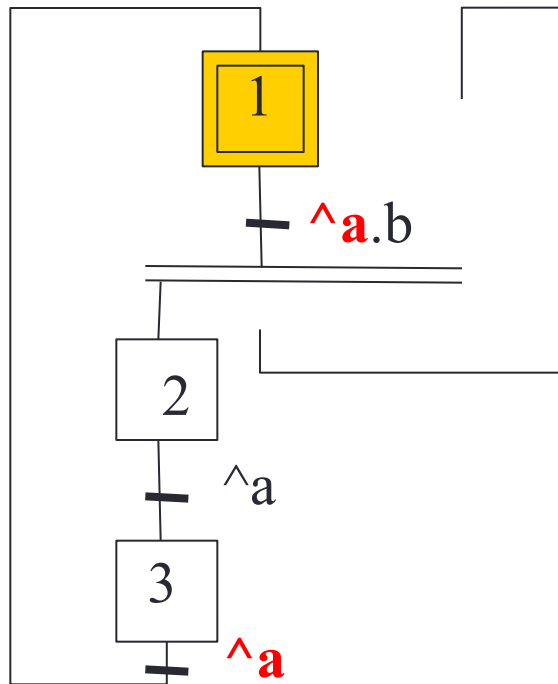
Etape simultanément activée et désactivée



Etape simultanément activée et désactivée

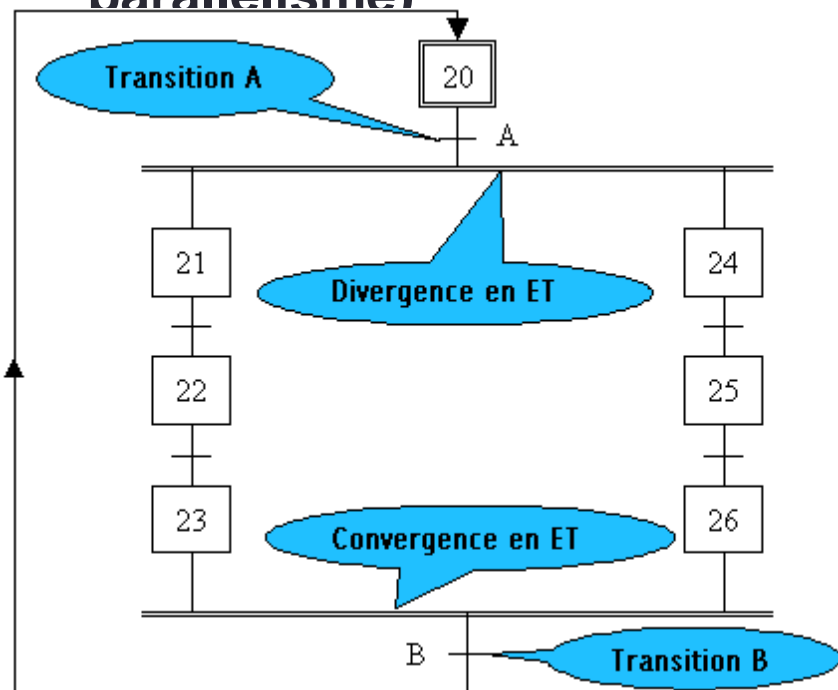


Etape simultanément activée et désactivée



STRUCTURES DE BASE

Divergence et convergence en ET (séquences simultanées ou parallélisme)



Divergence en ET : lorsque la transition A est franchie, les étapes 21 et 24 sont actives.

Convergence en ET : la transition B sera validée lorsque les étapes 23 et 26 seront actives. Si la réceptivité associée à cette transition est vraie, alors celle-ci est franchie.

REMARQUES :

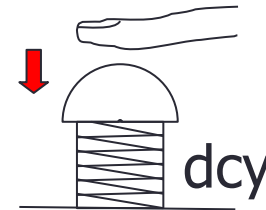
Après une divergence en ET, on trouve généralement une convergence en ET. Le nombre de branches parallèles peut-être supérieur à 2.

La réceptivité associée à la convergence peut-être de la forme $= 1$. Dans ce cas la transition est franchie dès qu'elle est active.

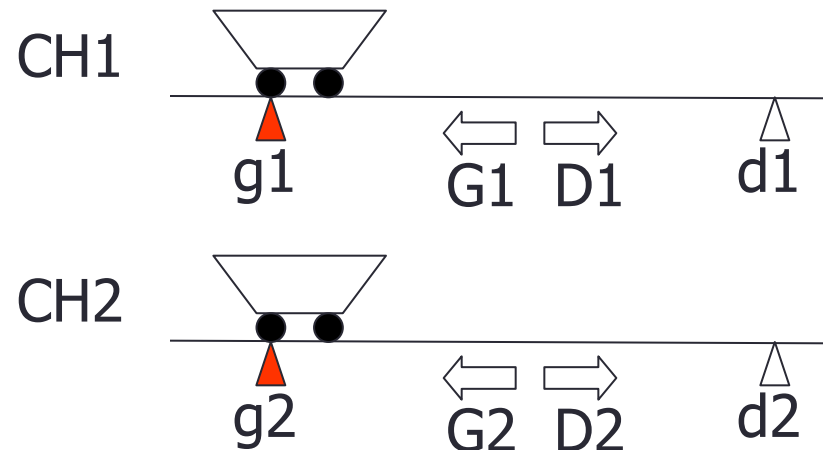
Exemple avec branchement ET (fonctionnement parallèle)

Cahier des charges :

après appui sur départ cycle « dcy », les chariots partent pour un aller-retour. Un nouveau départ cycle ne peut se faire que si les deux chariots sont à gauche.



CH1, CH2 : chariot 1, 2
 g : capteur « position gauche »
 d : capteur « position droite »
 G : action « aller à gauche »
 D : action « aller à droite »



CH1, CH2 : chariot 1, 2

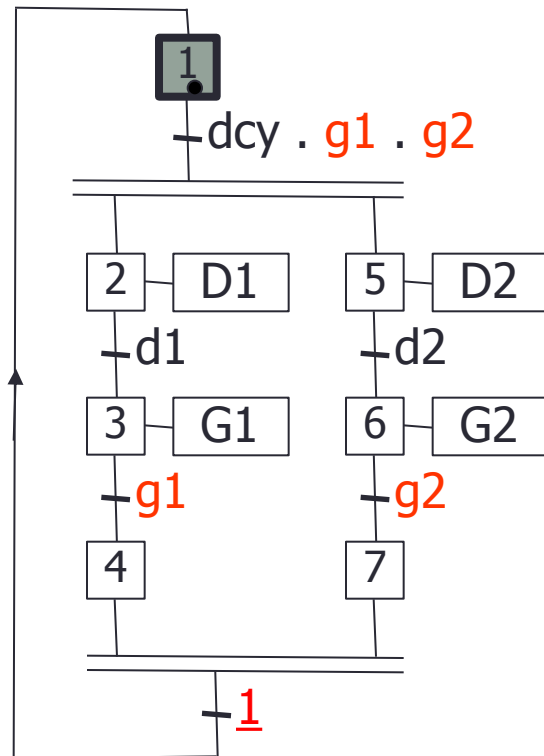
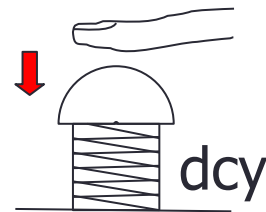
g : capteur « position gauche »

d : capteur « position droite »

G : action « aller à gauche »

D : action « aller à droite »

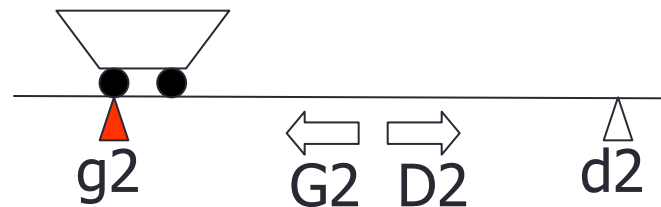
Solution 1



CH1



CH2



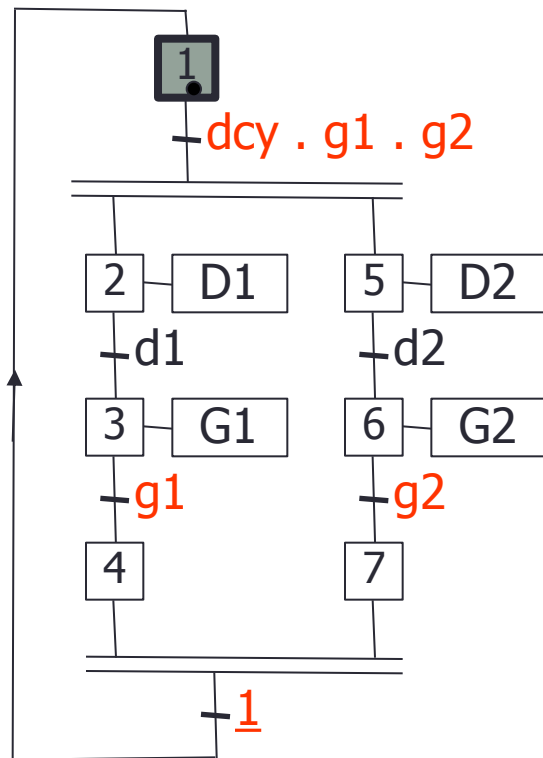
CH1, CH2 : chariot 1, 2

g : capteur « position gauche »

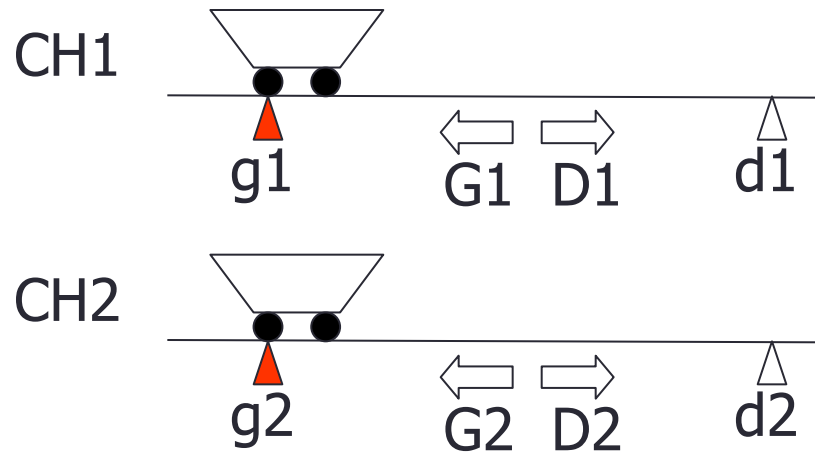
d : capteur « position droite »

G : action « aller à gauche »

D : action « aller à droite »



Solution 1



CH1, CH2 : chariot 1, 2

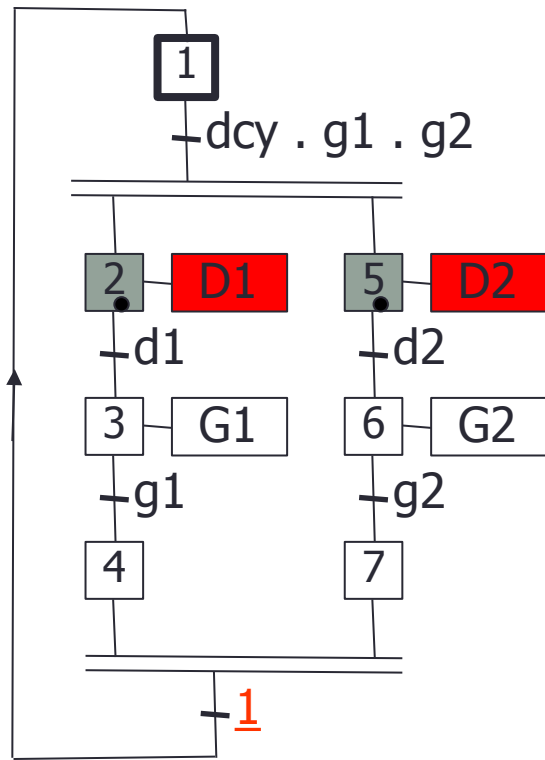
g : capteur « position gauche »

d : capteur « position droite »

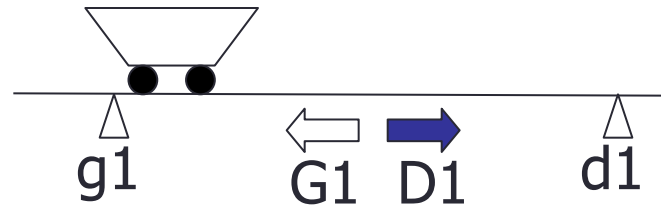
G : action « aller à gauche »

D : action « aller à droite »

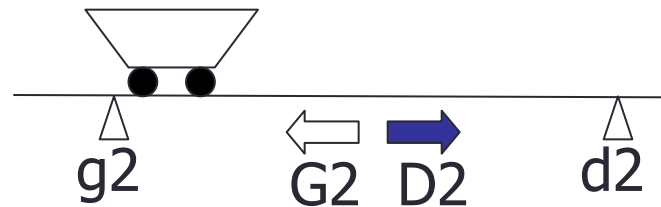
Solution 1



CH1



CH2



CH1, CH2 : chariot 1, 2

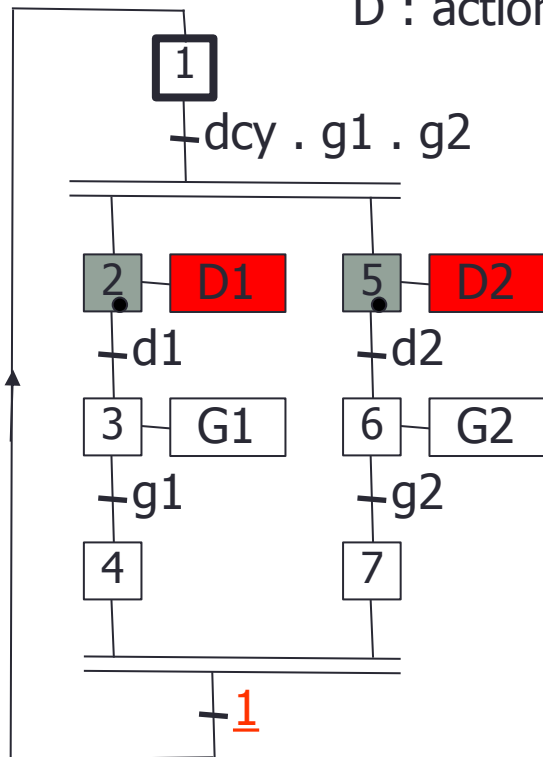
g : capteur « position gauche »

d : capteur « position droite »

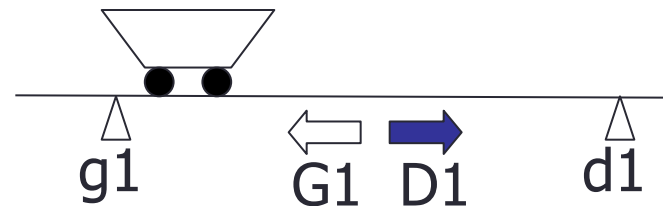
G : action « aller à gauche »

D : action « aller à droite »

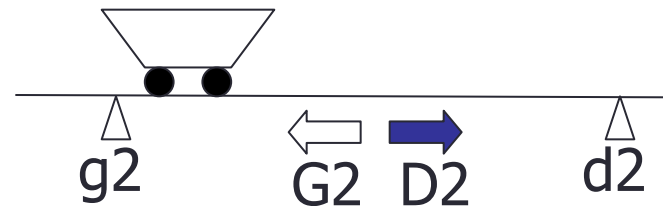
Solution 1



CH1



CH2



CH1, CH2 : chariot 1, 2

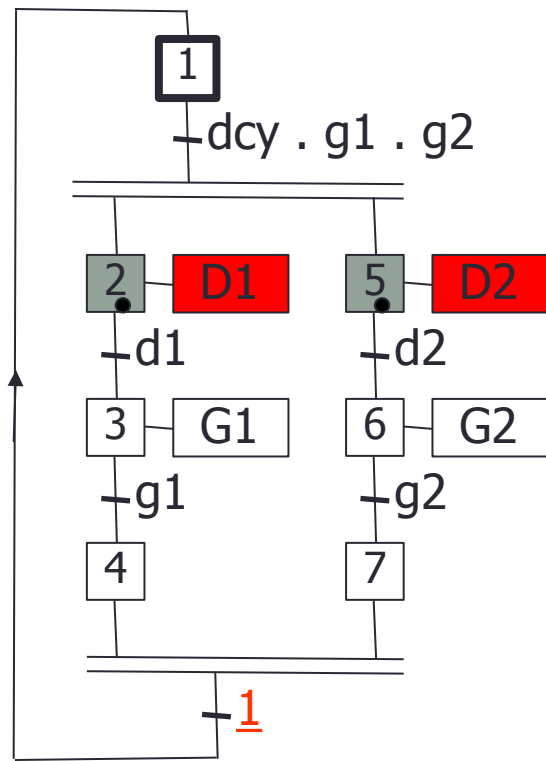
g : capteur « position gauche »

d : capteur « position droite »

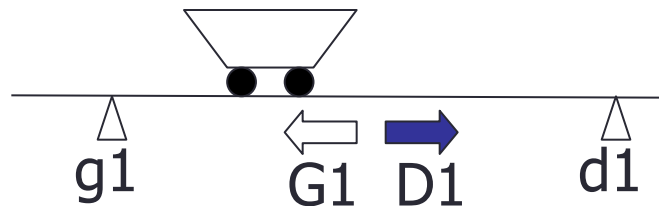
G : action « aller à gauche »

D : action « aller à droite »

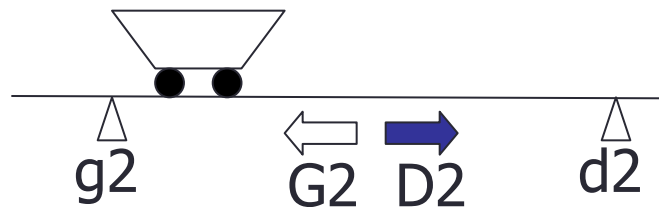
Solution 1



CH1



CH2



CH1, CH2 : chariot 1, 2

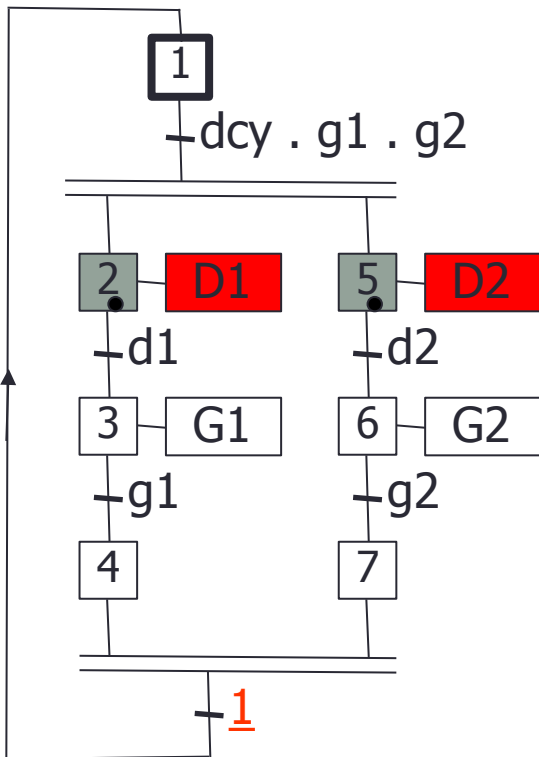
g : capteur « position gauche »

d : capteur « position droite »

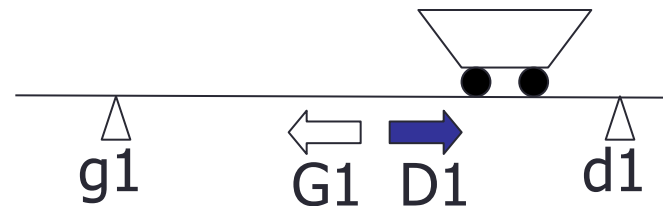
G : action « aller à gauche »

D : action « aller à droite »

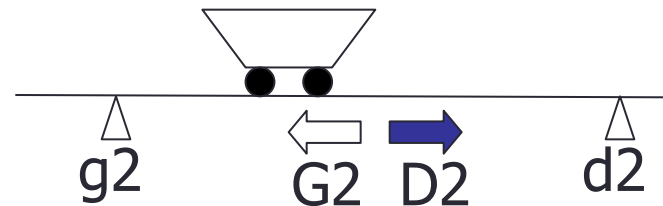
Solution 1



CH1



CH2



CH1, CH2 : chariot 1, 2

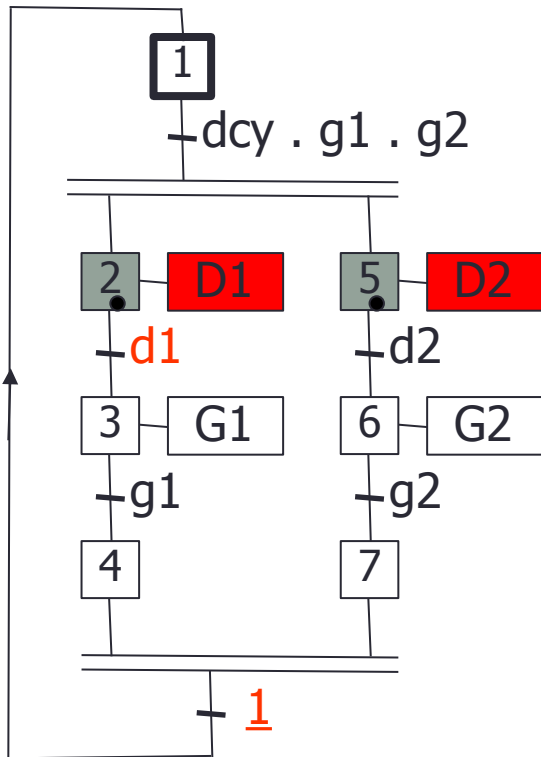
g : capteur « position gauche »

d : capteur « position droite »

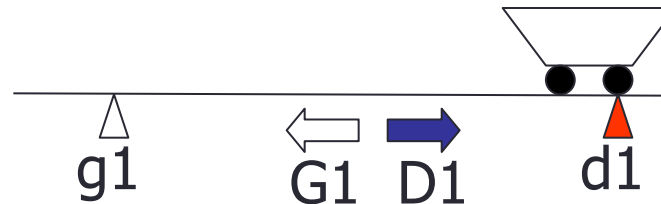
G : action « aller à gauche »

D : action « aller à droite »

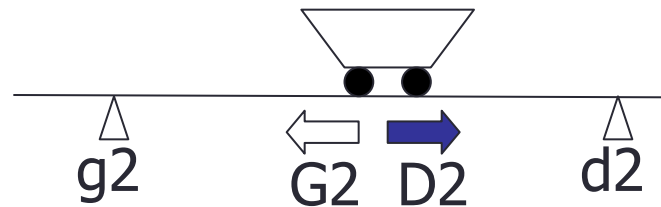
Solution 1



CH1



CH2



CH1, CH2 : chariot 1, 2

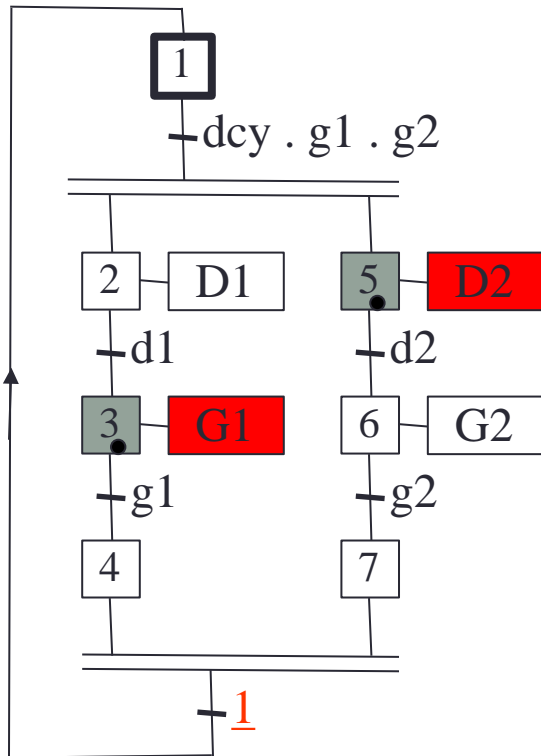
g : capteur « position gauche »

d : capteur « position droite »

G : action « aller à gauche »

D : action « aller à droite »

Solution 1



CH1



CH2



CH1, CH2 : chariot 1, 2

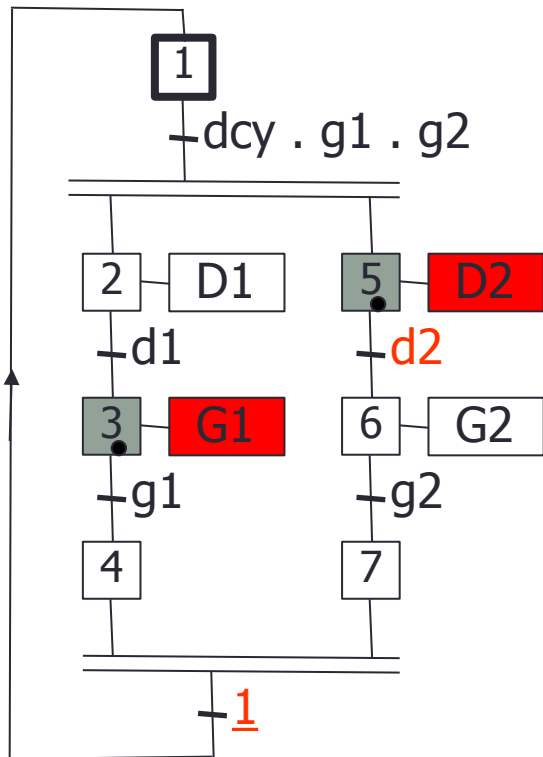
g : capteur « position gauche »

d : capteur « position droite »

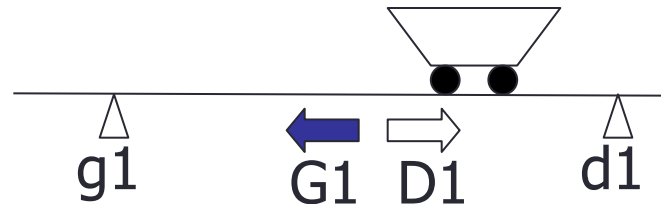
G : action « aller à gauche »

D : action « aller à droite »

Solution 1



CH1



CH2



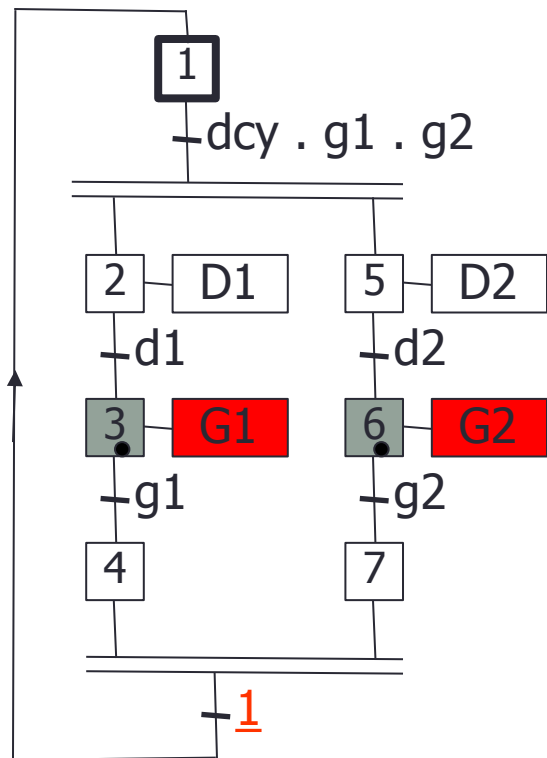
CH1, CH2 : chariot 1, 2

g : capteur « position gauche »

d : capteur « position droite »

G : action « aller à gauche »

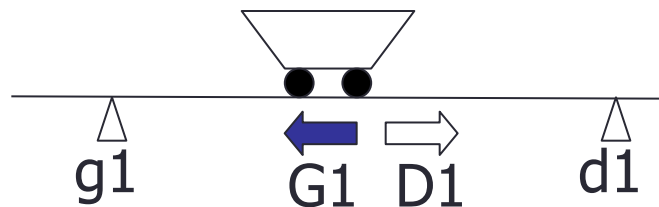
D : action « aller à droite »



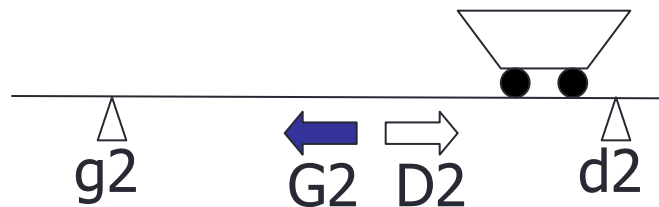
Solution 1



CH1



CH2



CH1, CH2 : chariot 1, 2

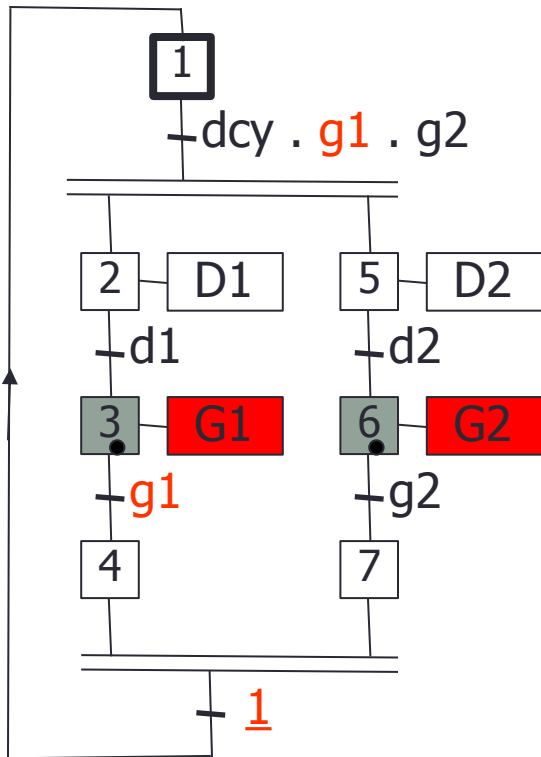
g : capteur « position gauche »

d : capteur « position droite »

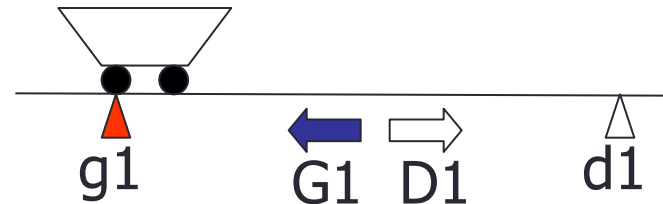
G : action « aller à gauche »

D : action « aller à droite »

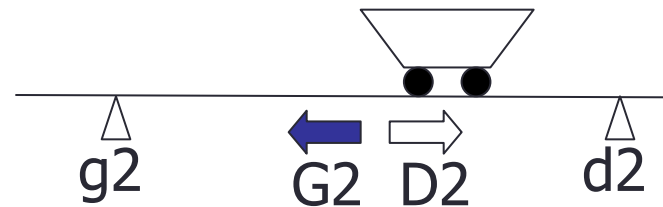
Solution 1



CH1



CH2



CH1, CH2 : chariot 1, 2

g : capteur « position gauche »

d : capteur « position droite »

G : action « aller à gauche »

D : action « aller à droite »

Solution 1

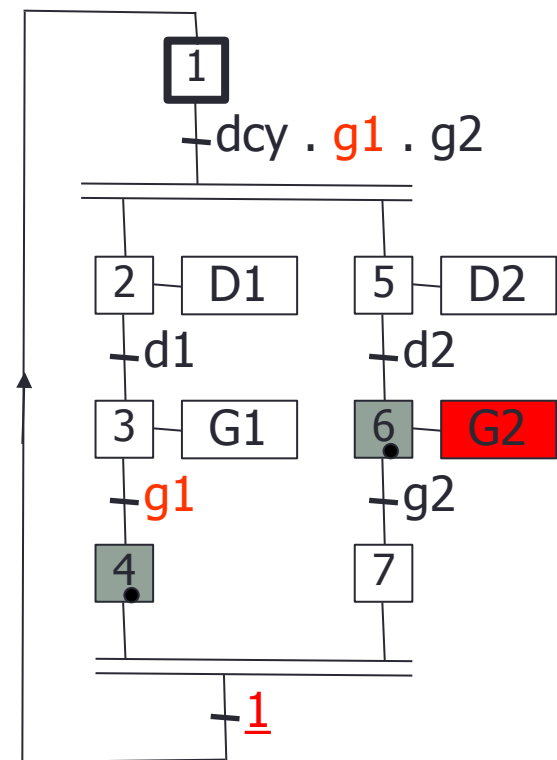
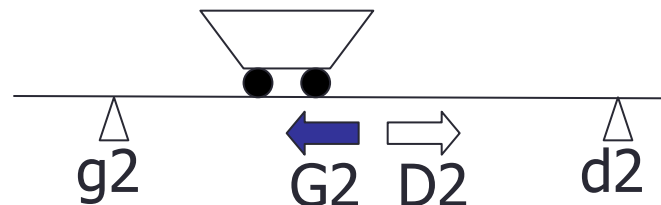


Etape 4 = étape « d'attente » ⇒ Aucune action

CH1



CH2



CH1, CH2 : chariot 1, 2

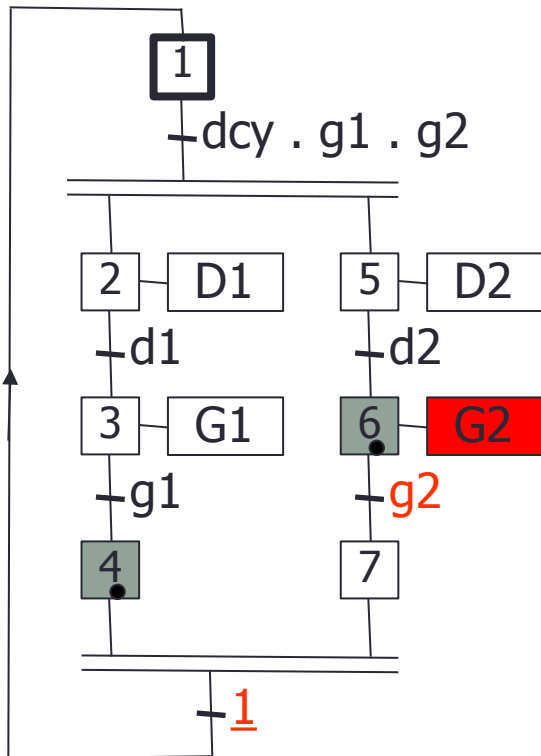
g : capteur « position gauche »

d : capteur « position droite »

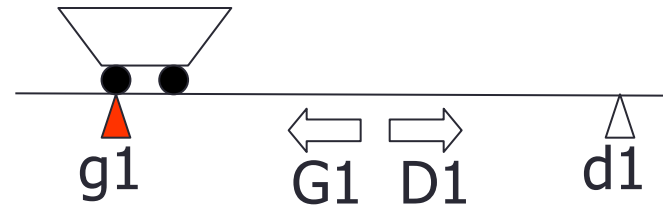
G : action « aller à gauche »

D : action « aller à droite »

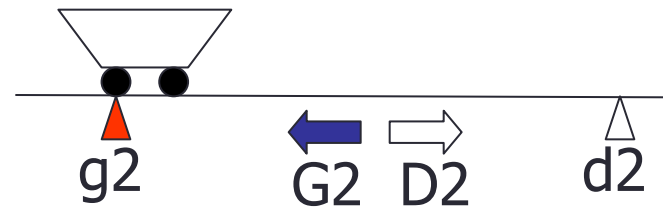
Solution 1



CH1



CH2



CH1, CH2 : chariot 1, 2

g : capteur « position gauche »

d : capteur « position droite »

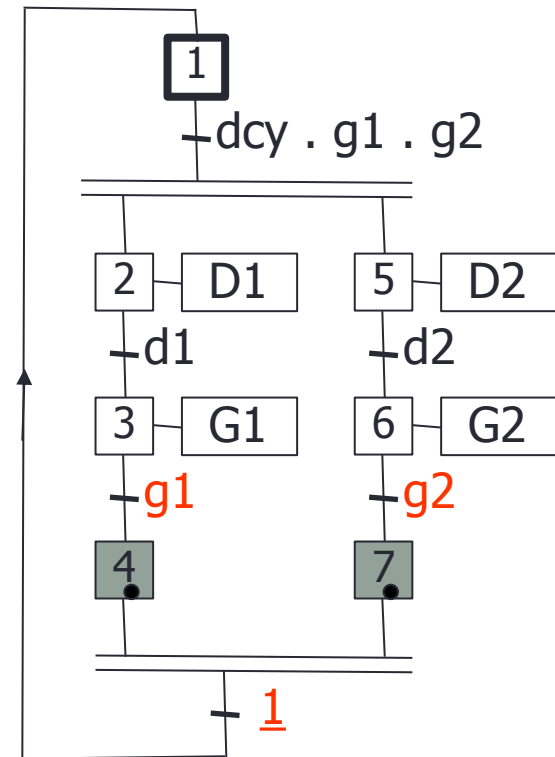
G : action « aller à gauche »

D : action « aller à droite »

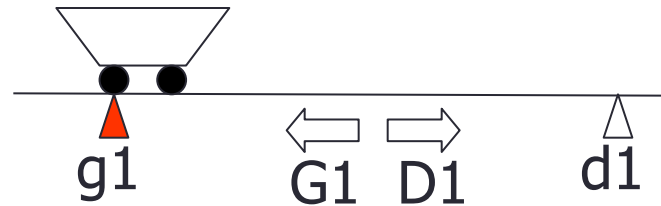
Solution 1



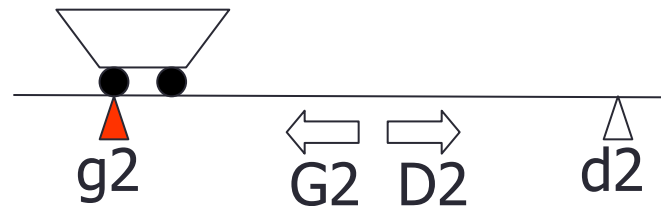
Étapes 4 & 7 actives ⇒ Synchronisation



CH1



CH2



CH1, CH2 : chariot 1, 2

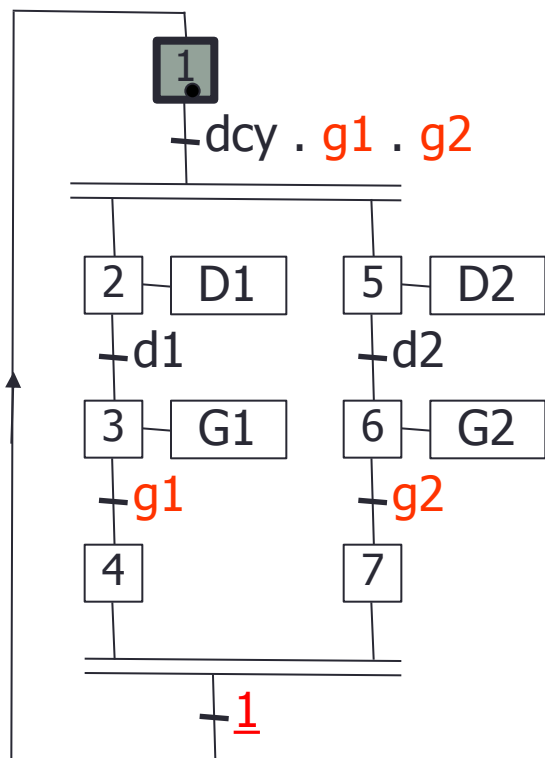
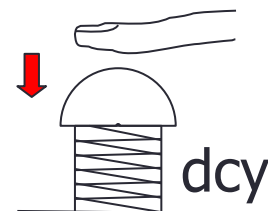
g : capteur « position gauche »

d : capteur « position droite »

G : action « aller à gauche »

D : action « aller à droite »

Solution 1



CH1



CH2



Autre solution

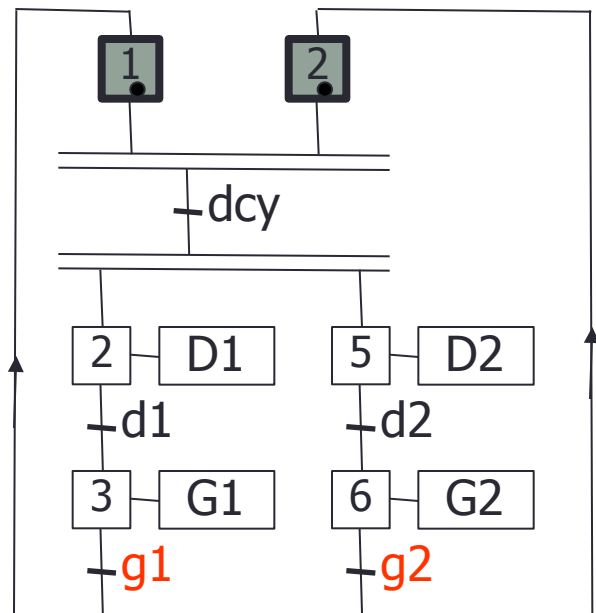
CH1, CH2 : chariot 1, 2

g : capteur « position gauche »

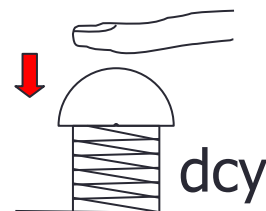
d : capteur « position droite »

G : action « aller à gauche »

D : action « aller à droite »



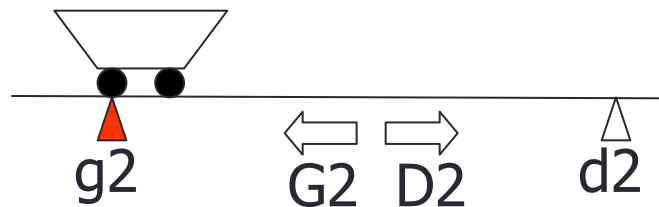
Solution 2

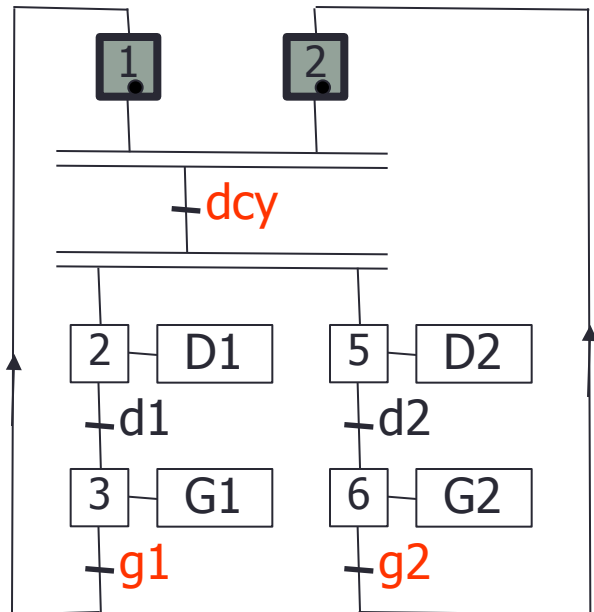


CH1

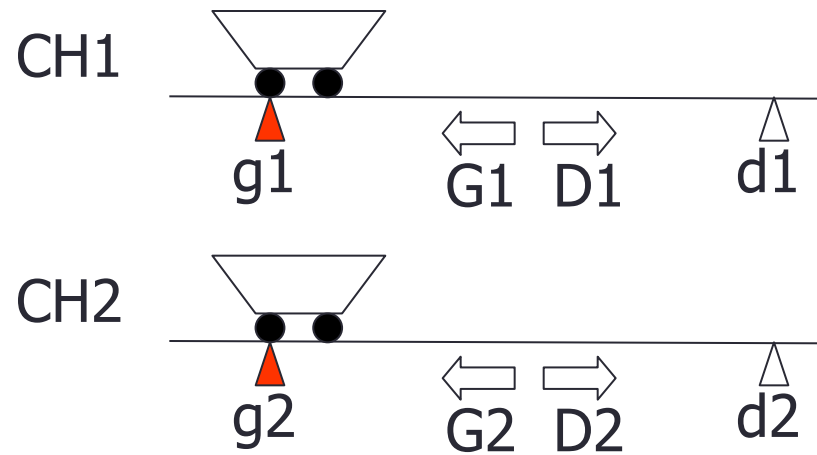


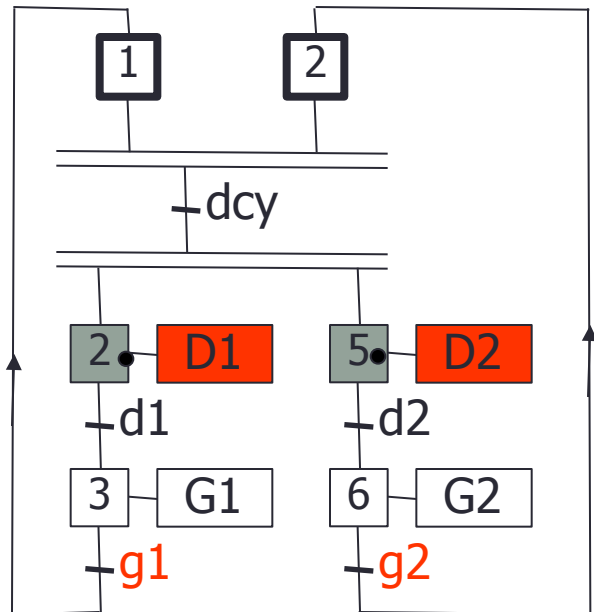
CH2



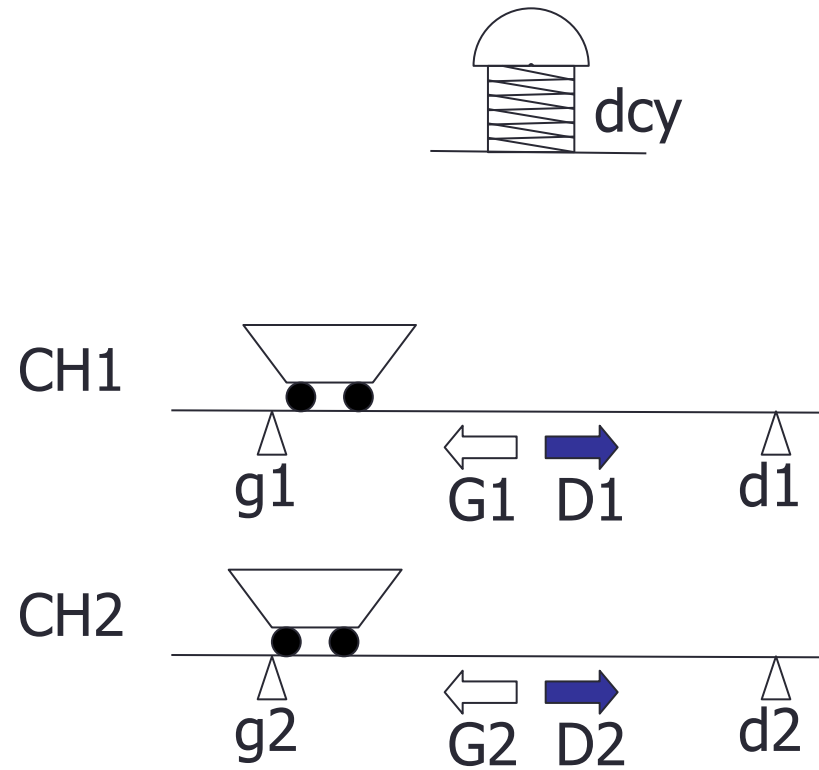


Solution 2

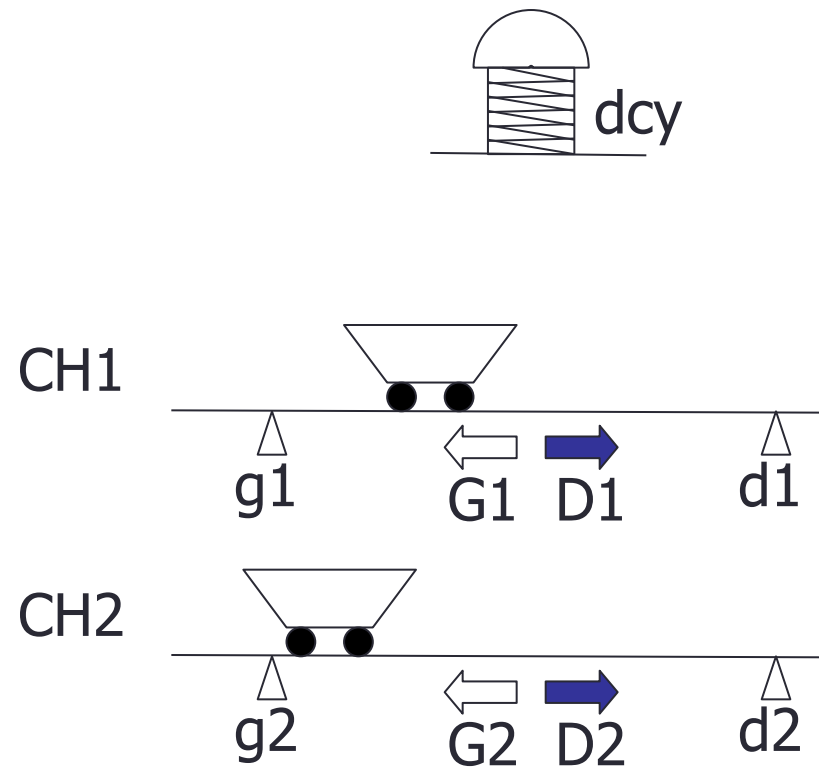
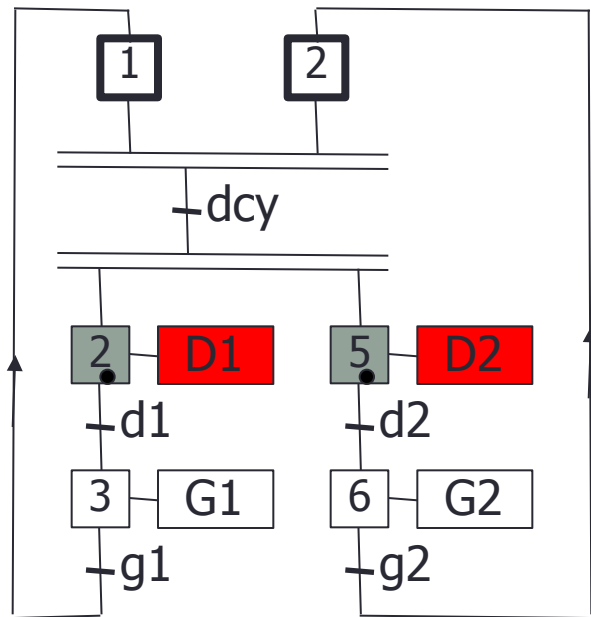


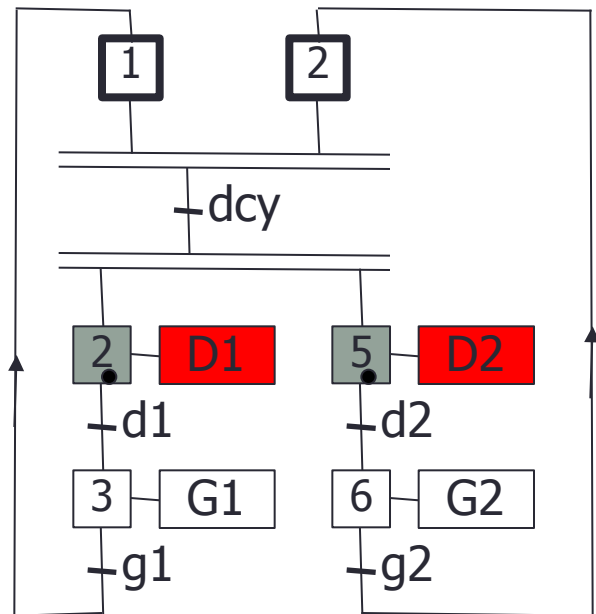


Solution 2

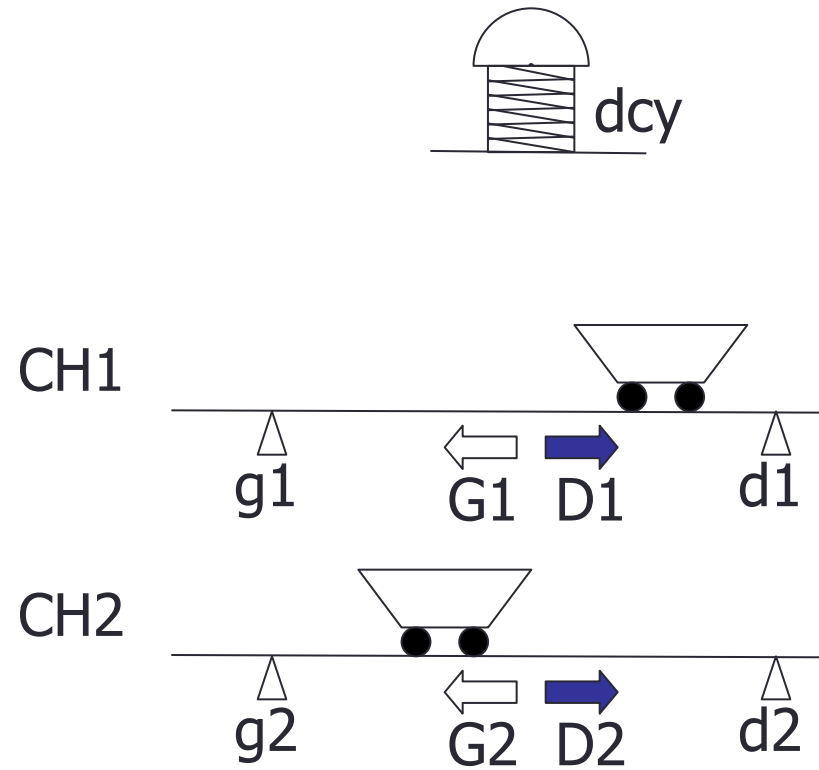


Solution 2

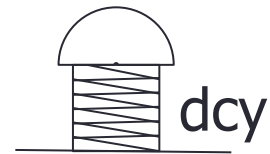
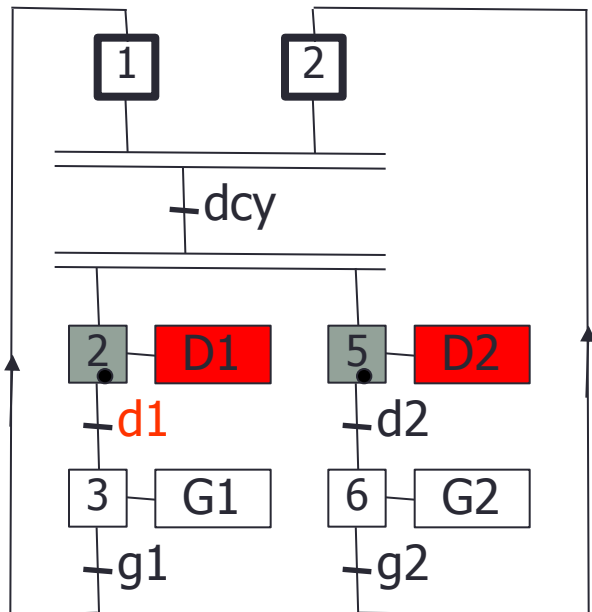




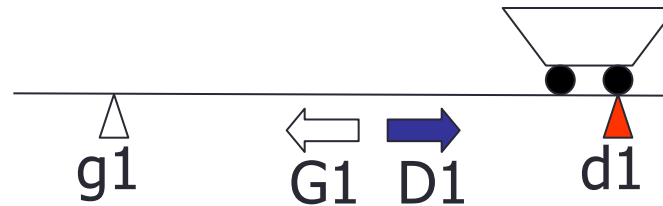
Solution 2



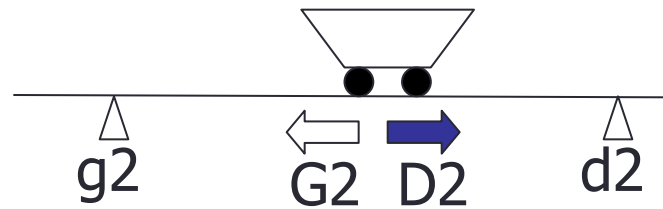
Solution 2

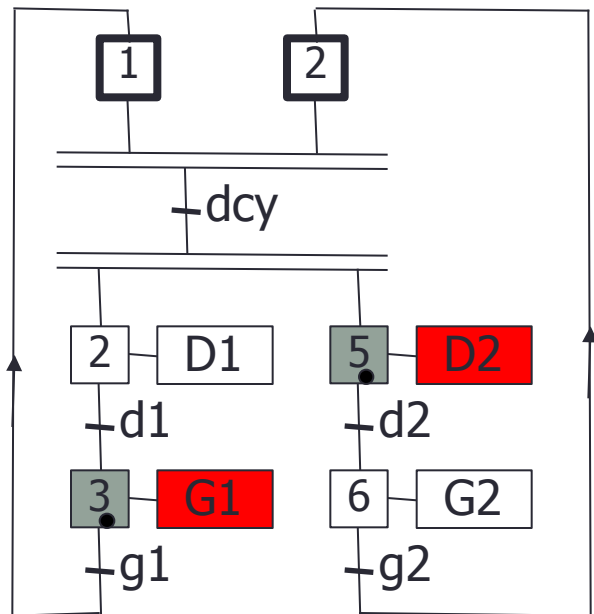


CH1



CH2

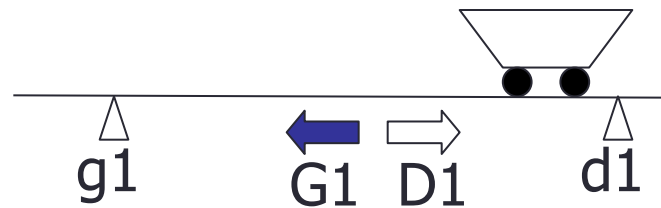




Solution 2

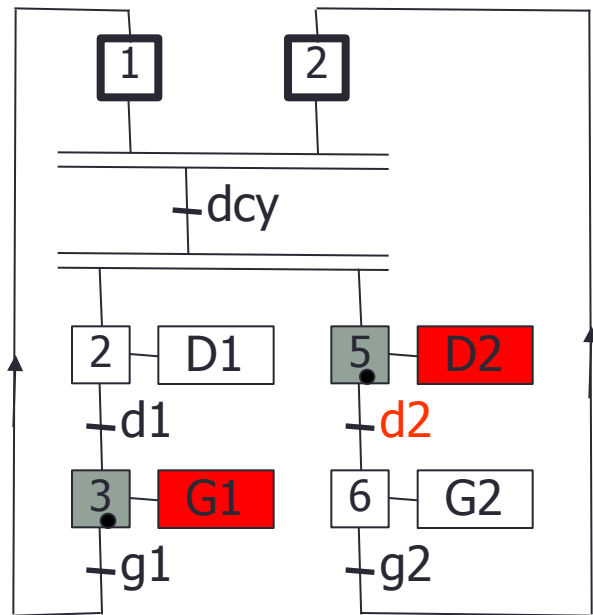


CH1

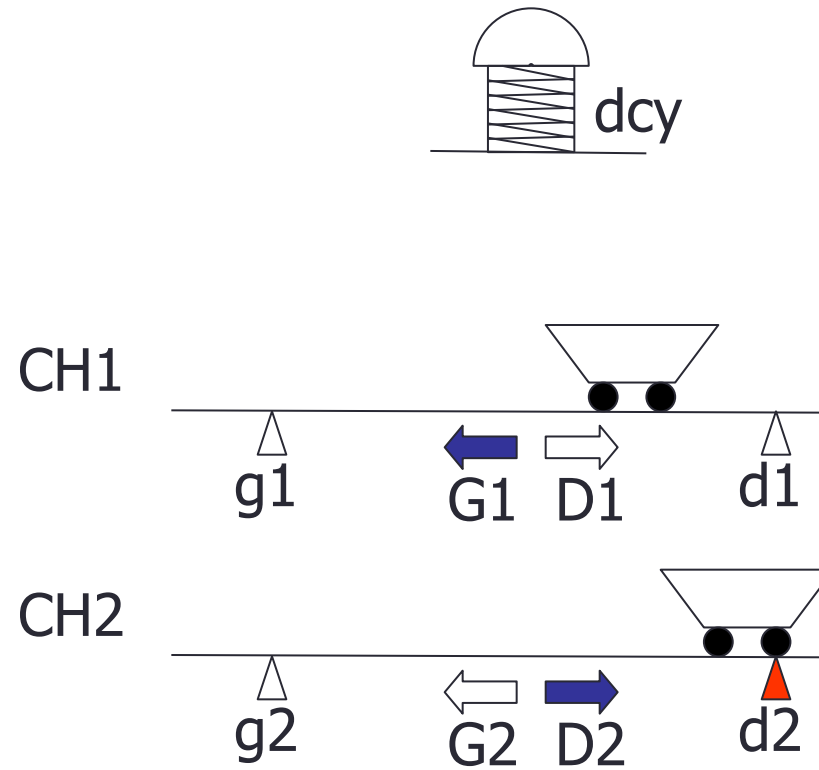


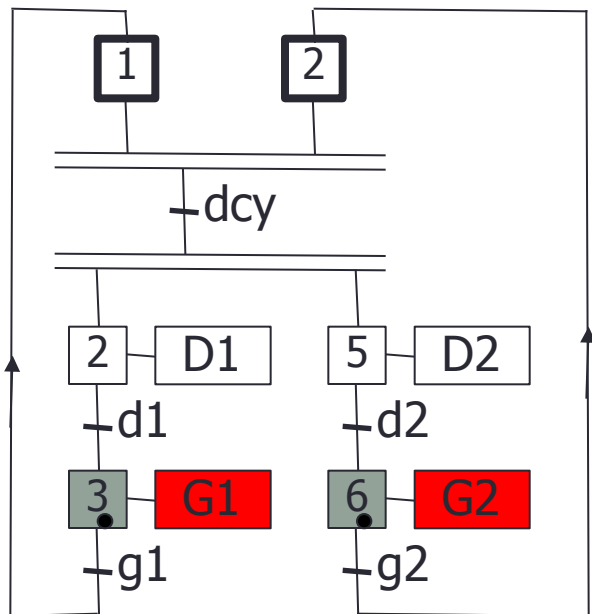
CH2



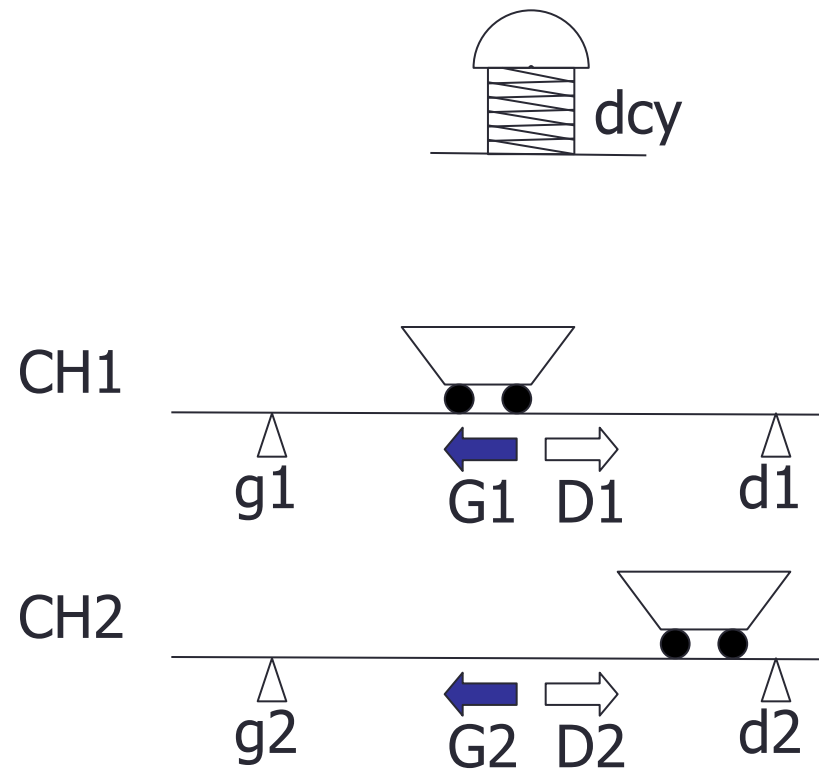


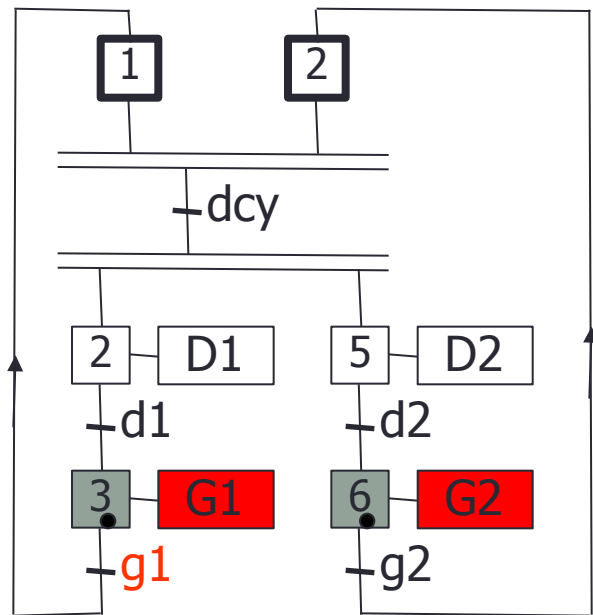
Solution 2



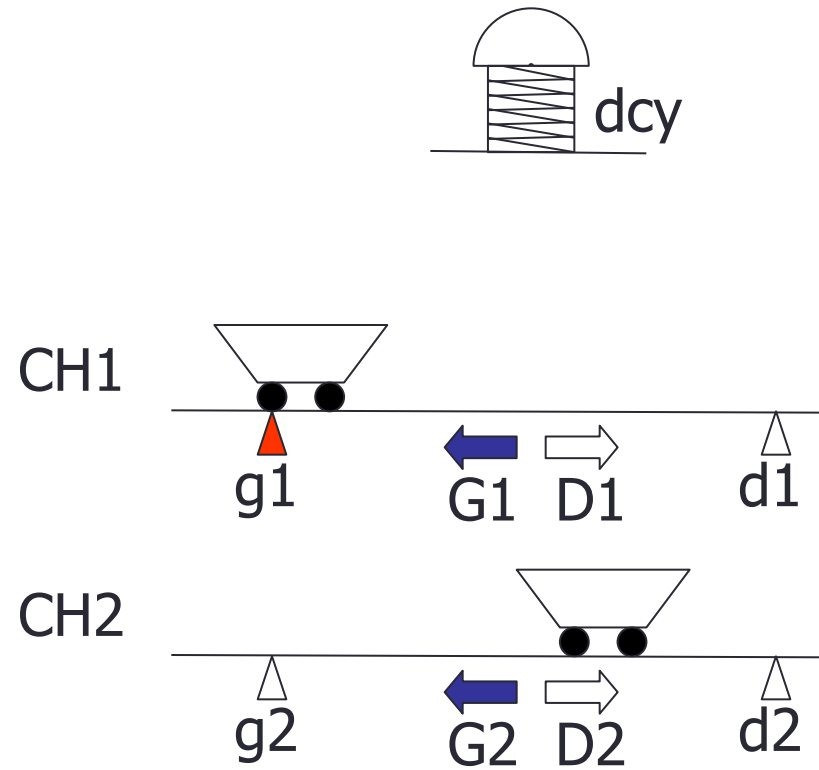


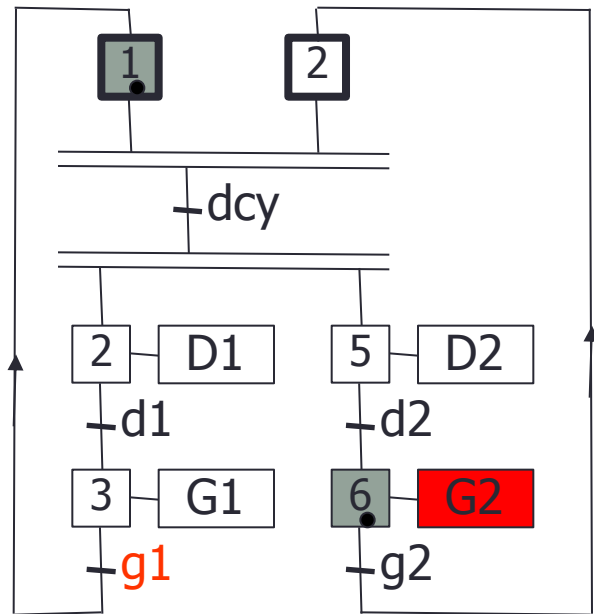
Solution 2



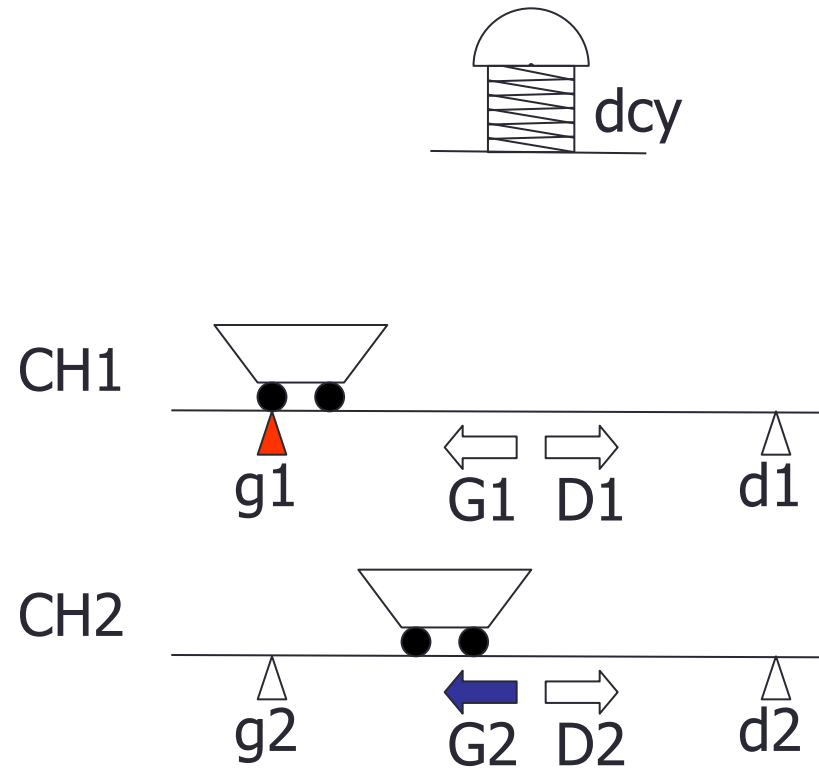


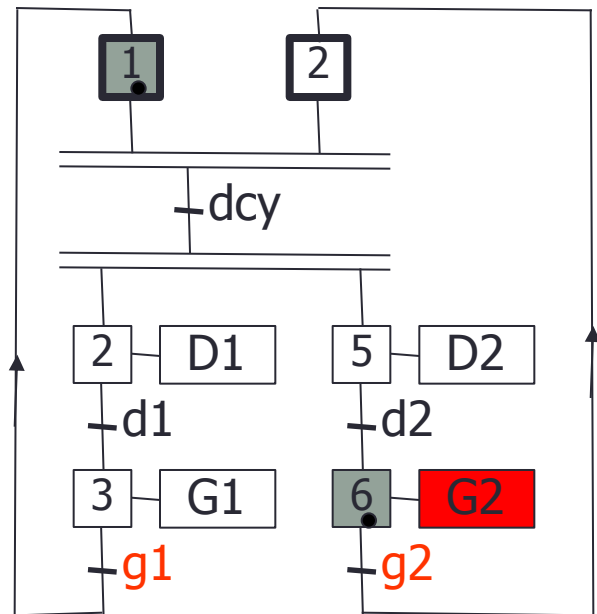
Solution 2



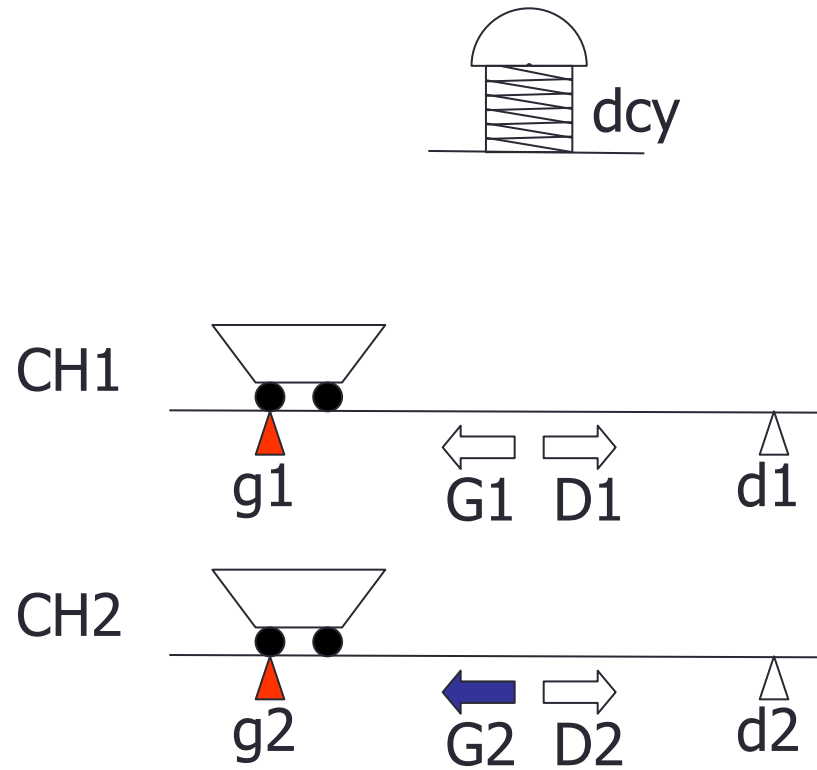


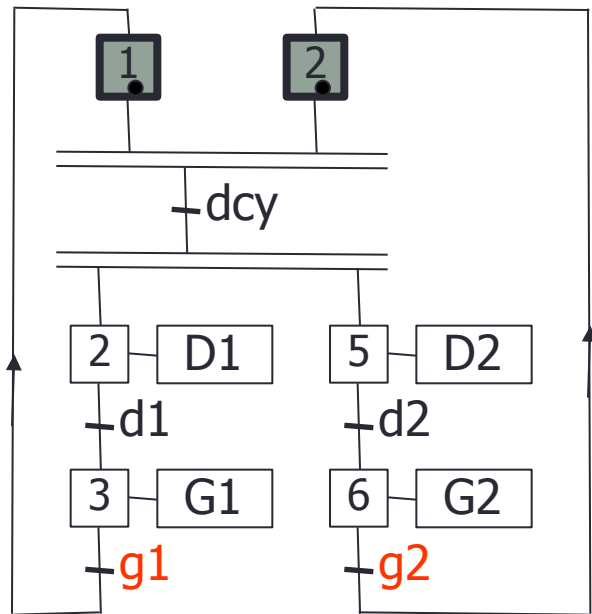
Solution 2



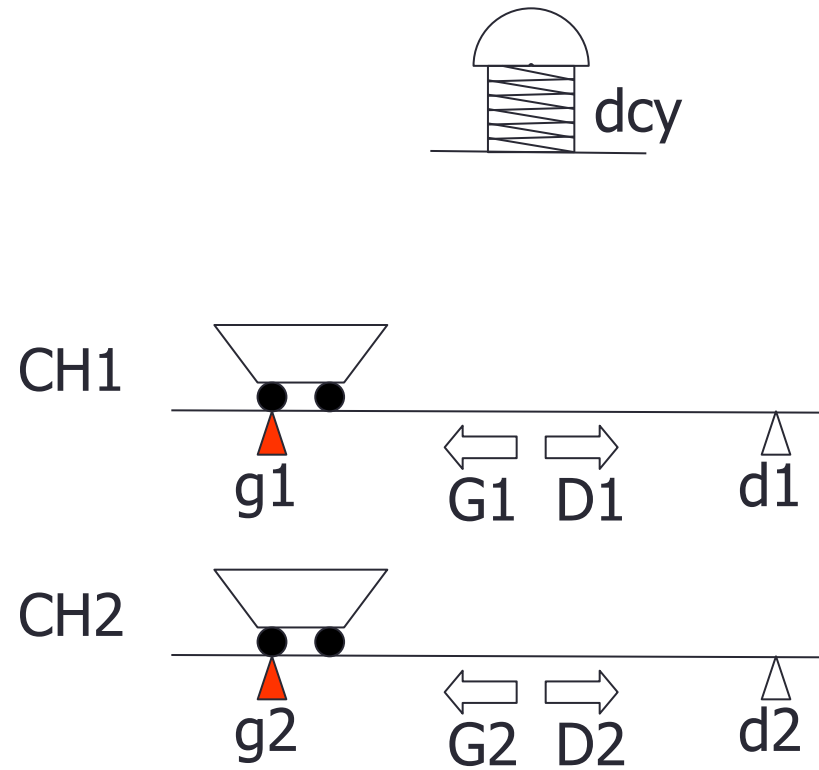


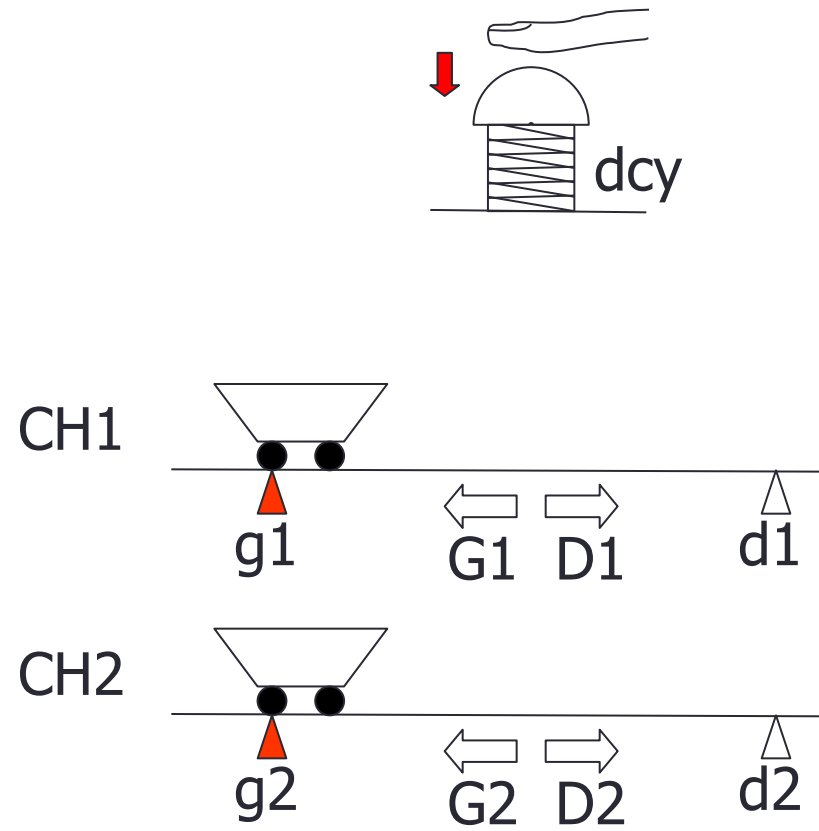
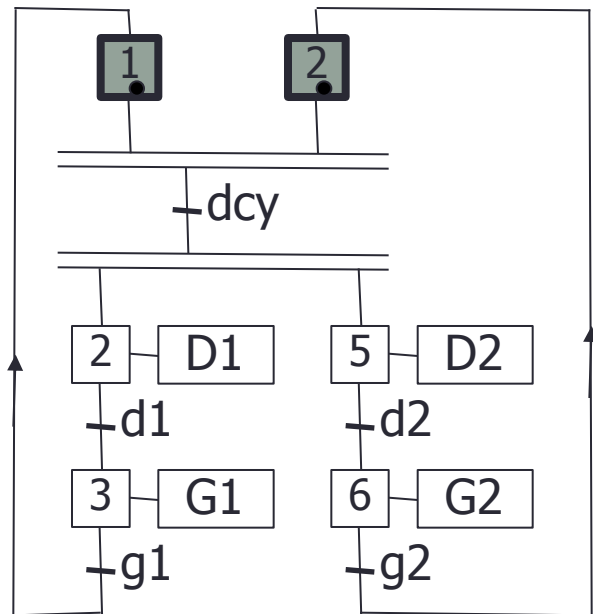
Solution 2



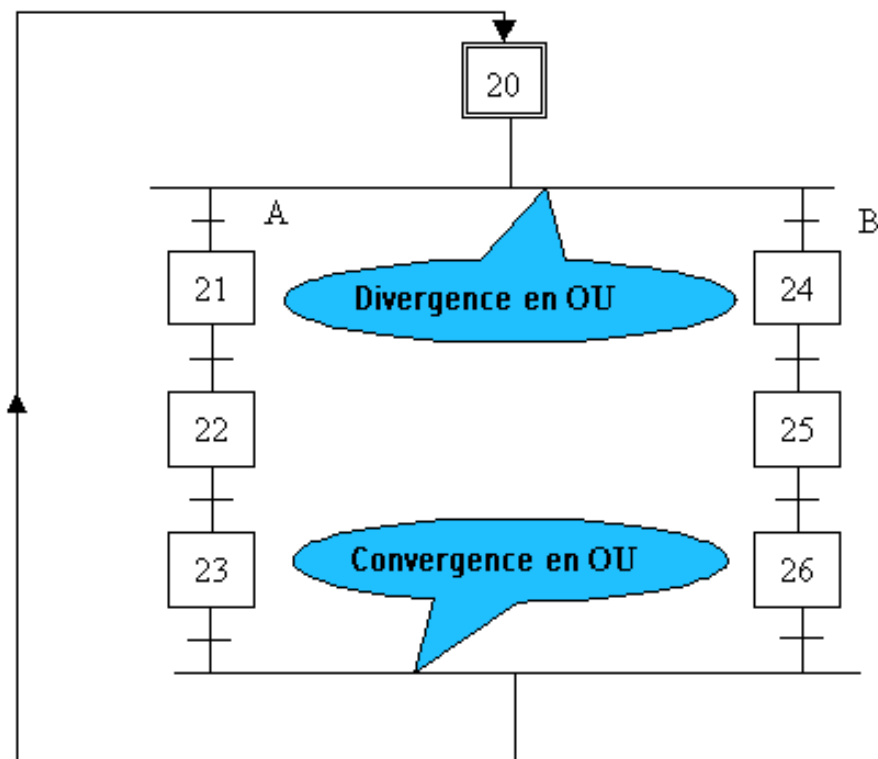


Solution 2





Divergence et convergence en OU (aiguillage)



Divergence en OU : l'évolution du système vers une branche dépend des réceptivités A et B associées aux transitions.

Convergence en OU : après l'évolution dans une branche, il y a convergence vers une étape commune.

REMARQUES :

A et B ne peuvent être vrais simultanément (conflit).

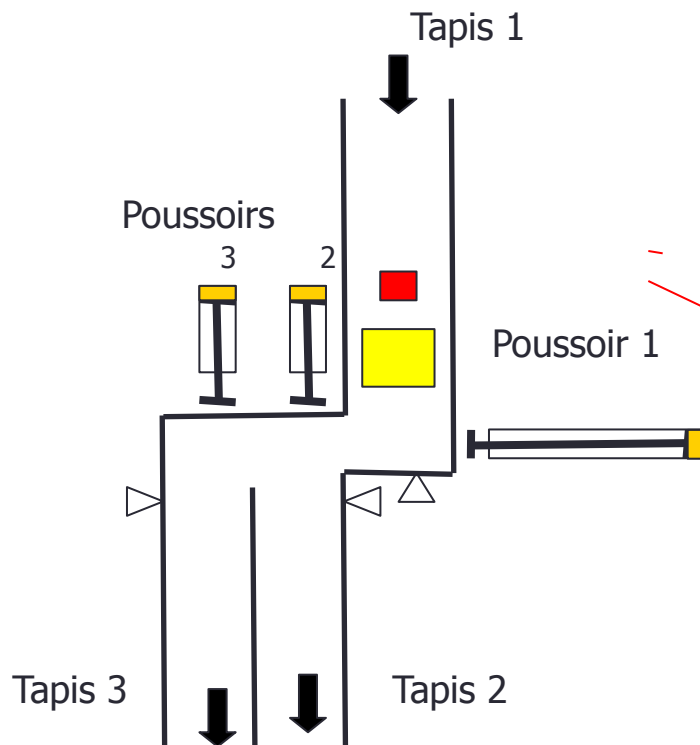
Après une divergence en OU, on trouve une convergence en OU.

Le nombre de branches peut-être supérieur à 2.

La convergence de toutes les branches ne se fait pas obligatoirement au même endroit.

Exemple avec branchement OU (sélection de séquences)

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 suivant la figure ci-dessous :

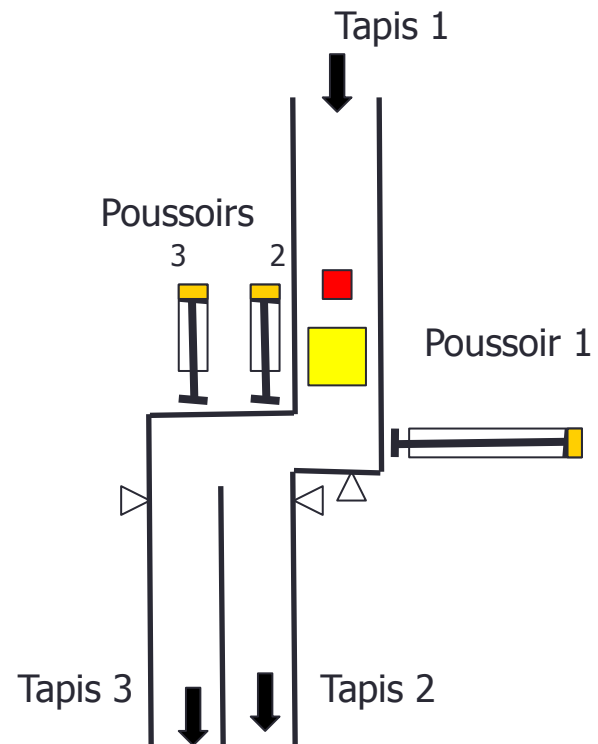
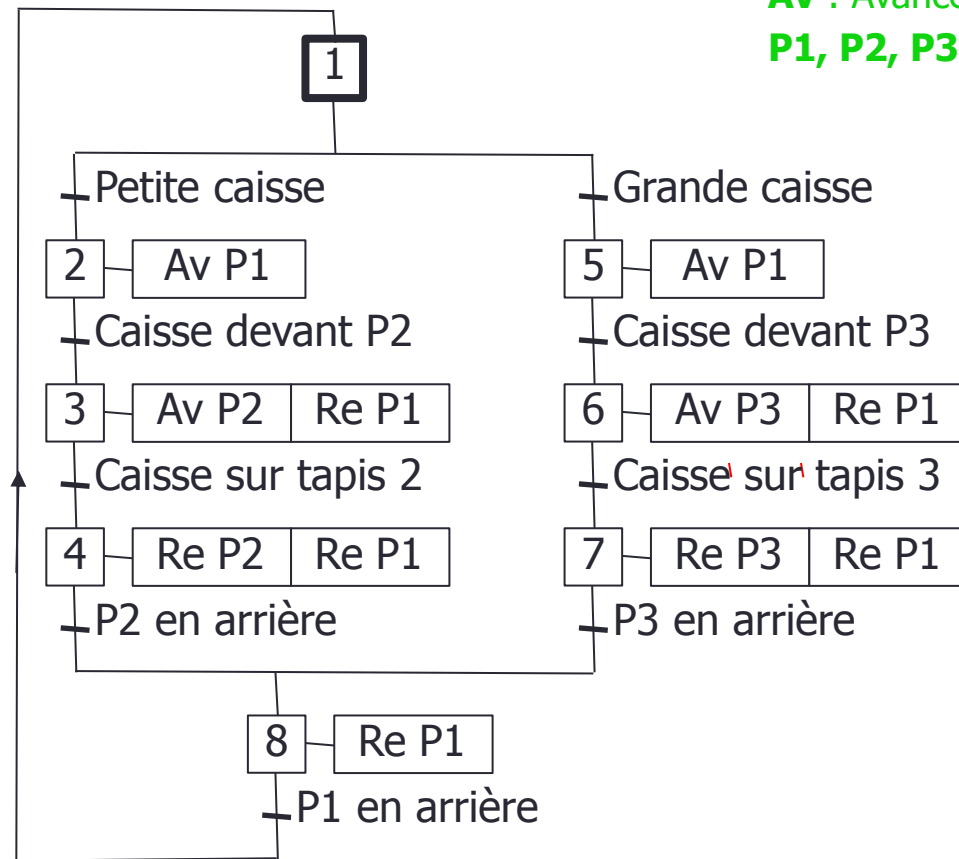


Cycle de fonctionnement :

Le poussoir 1 pousse les petites caisses devant le poussoir 2 qui, à son tour, les transfère sur le tapis d'évacuation 2, alors que les grandes caisses sont poussées devant le poussoir 3, ce dernier les évacuant sur le tapis 3. Pour effectuer la sélection des caisses, un dispositif de détection placé devant le poussoir 1 permet de reconnaître sans ambiguïté le type de caisse qui se présente.

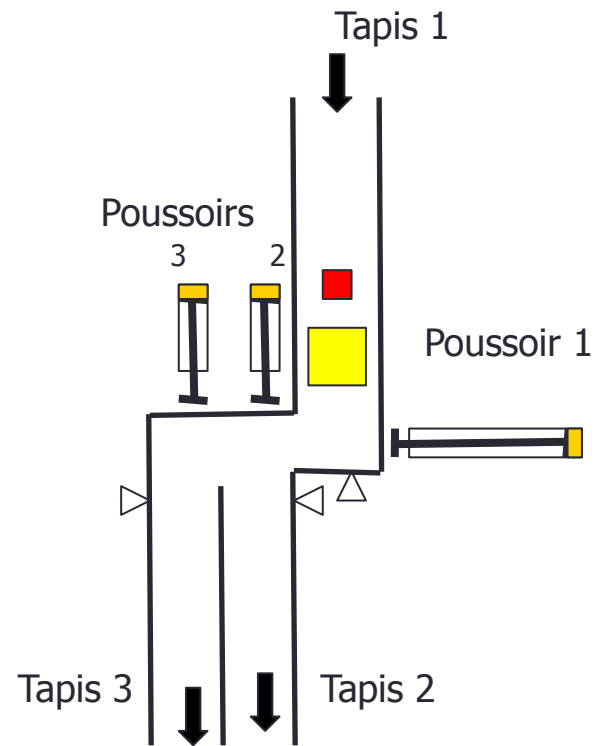
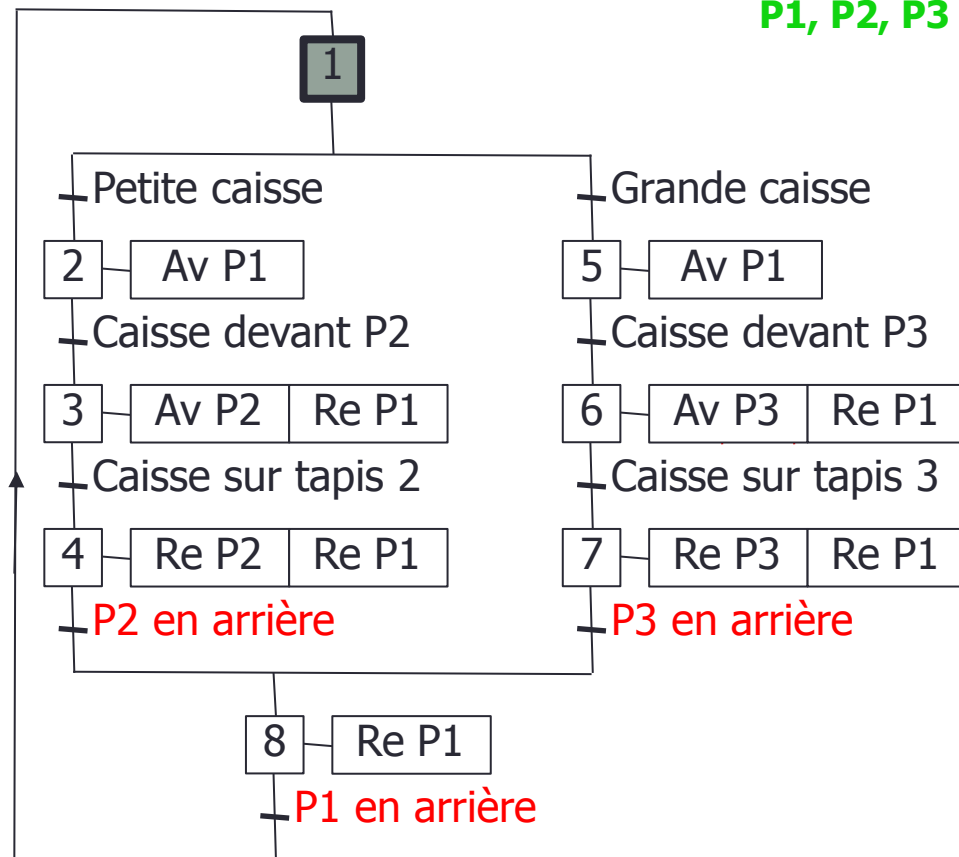
Exemple avec branchement OU (sélection de séquences)

Av : Avance **Re** : Recule
P1, P2, P3 : poussoirs 1, 2, 3



Exemple avec branchement OU (sélection de séquences)

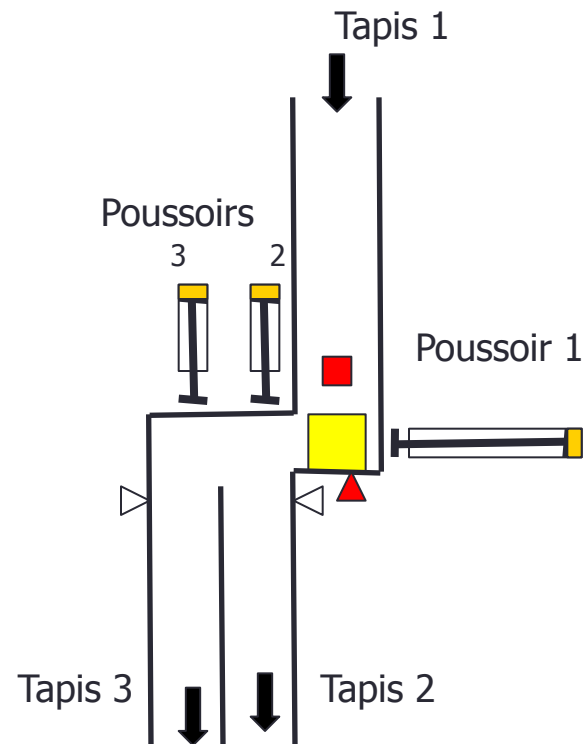
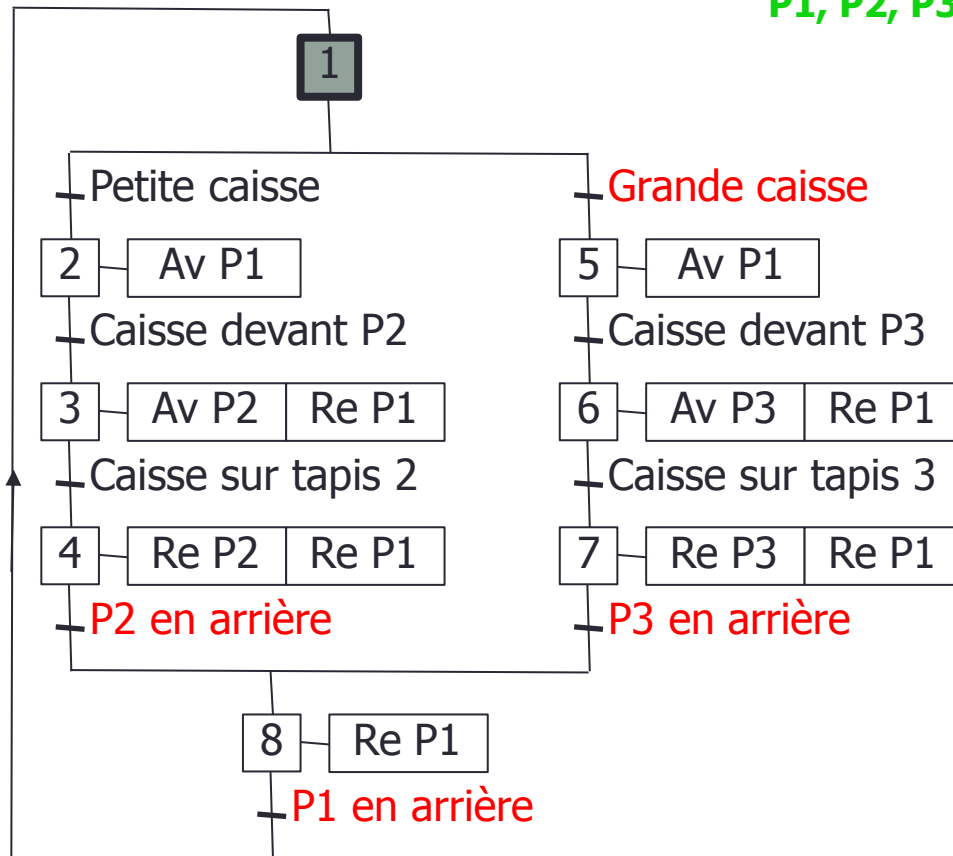
Av : Avance **Re** : Recule
P1, P2, P3 : poussoirs 1, 2, 3



Exemple avec branchement OU (sélection de séquences)

Av : Avance **Re** : Recule

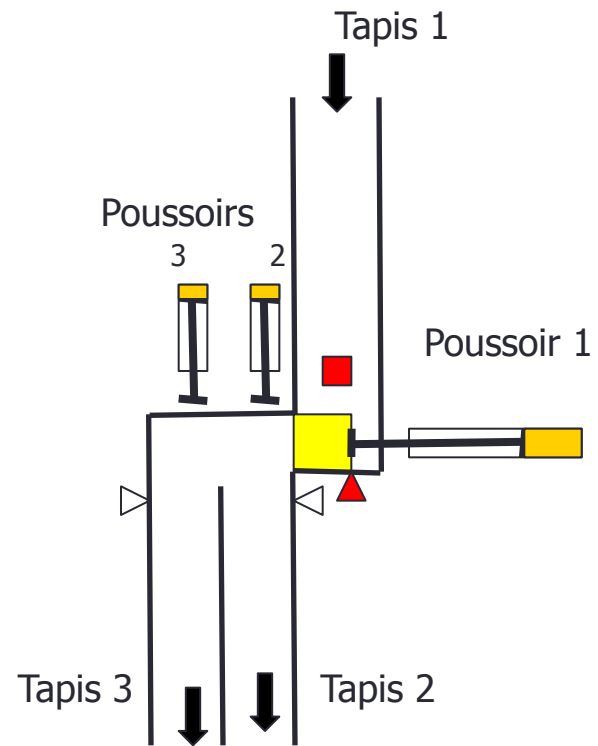
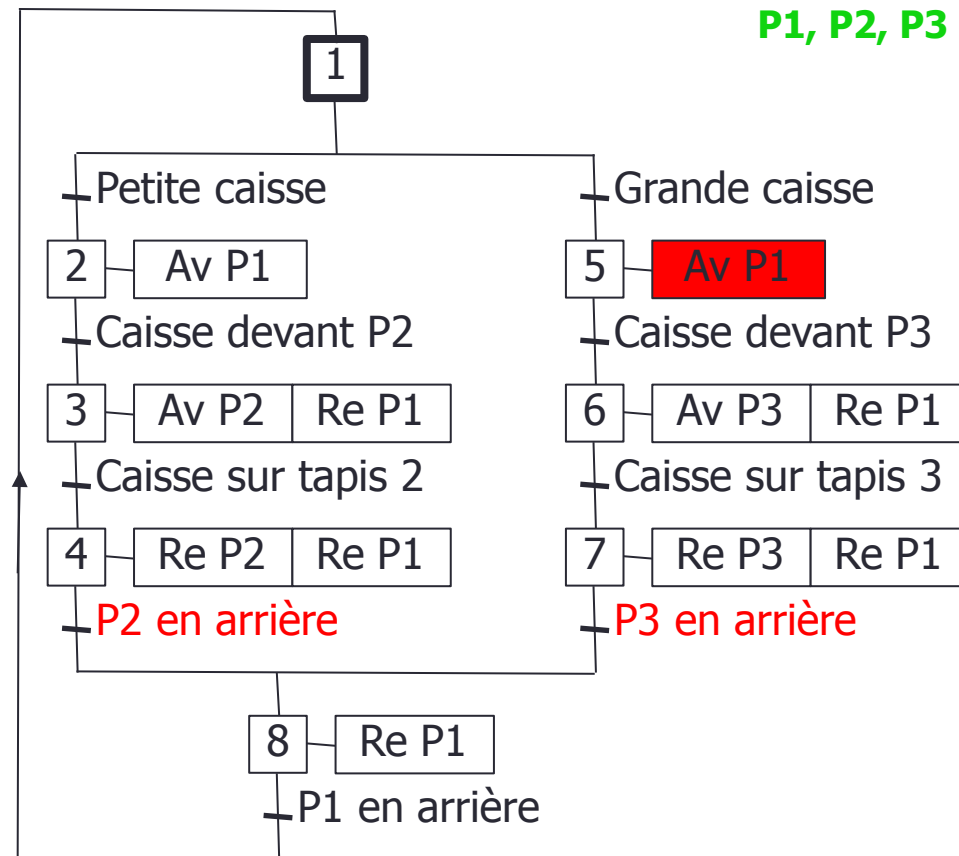
P1, P2, P3 : poussoirs 1, 2, 3



Exemple avec branchement OU (sélection de séquences)

Av : Avance **Re** : Recule

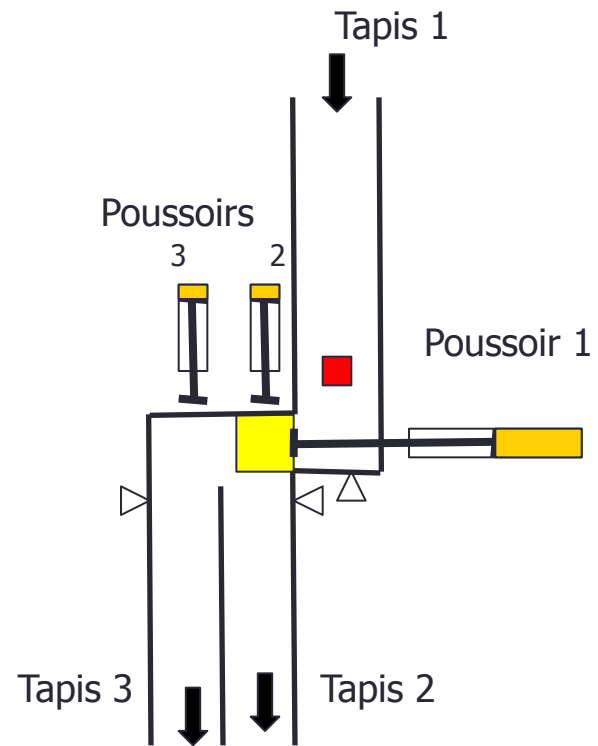
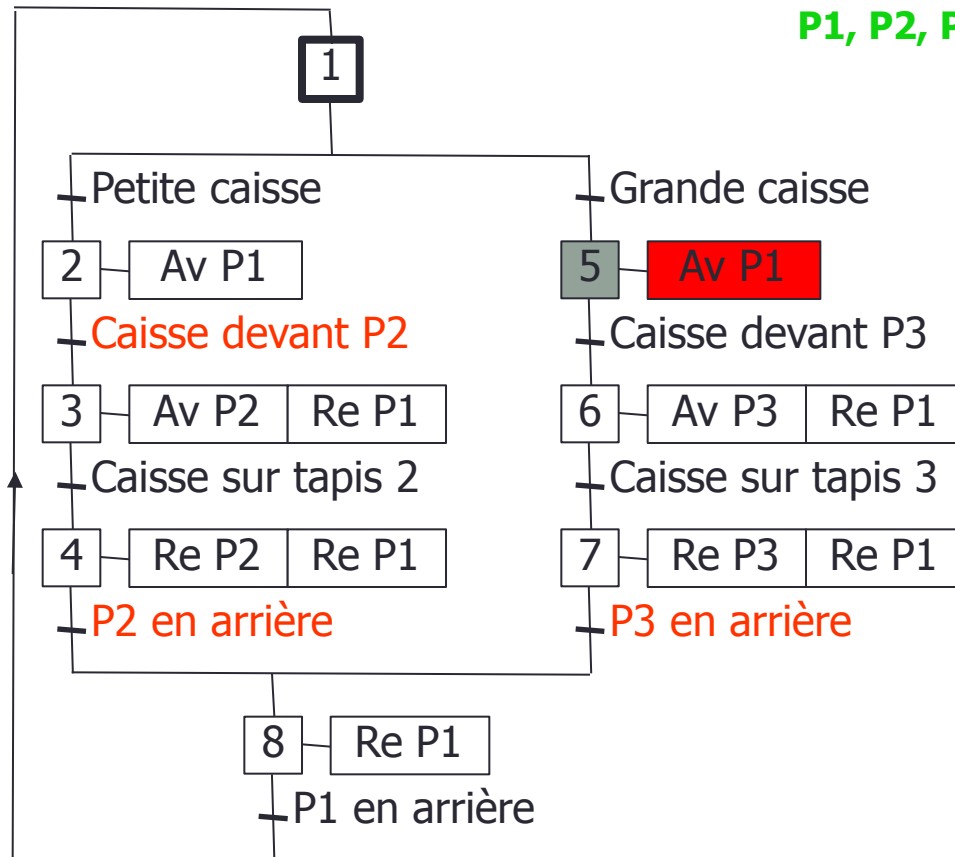
P1, P2, P3 : poussoirs 1, 2, 3



Exemple avec branchement OU (sélection de séquences)

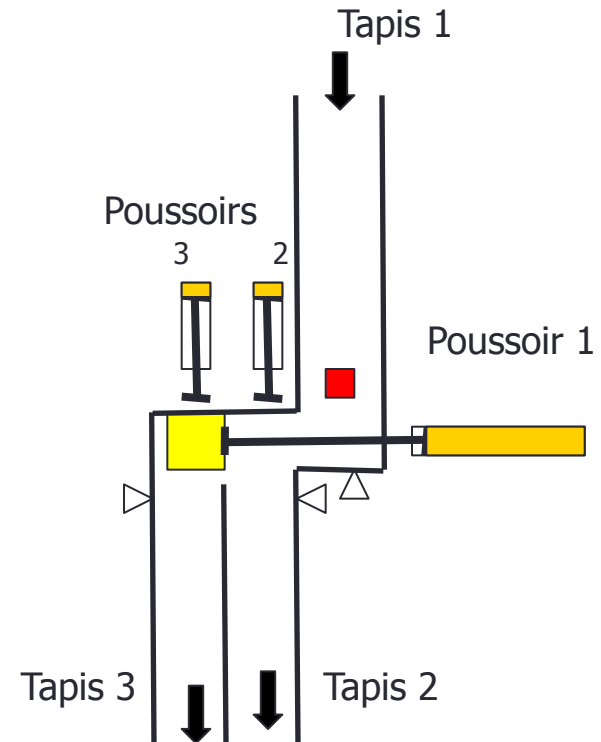
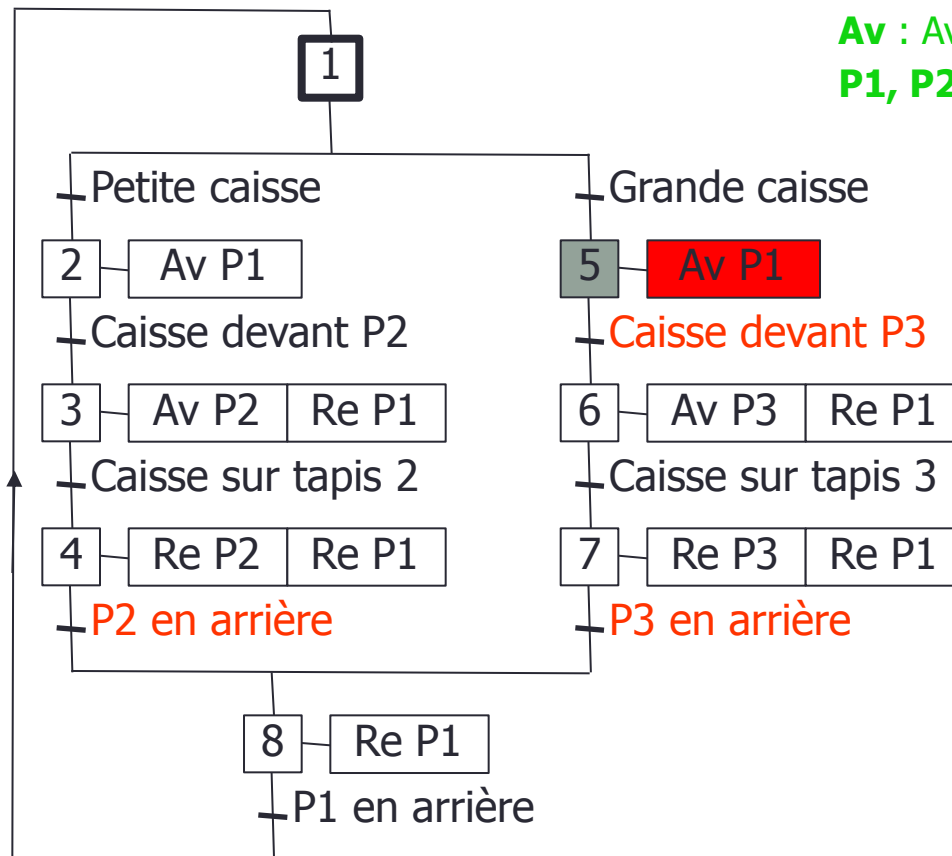
Av : Avance **Re** : Recule

P1, P2, P3 : poussoirs 1, 2, 3



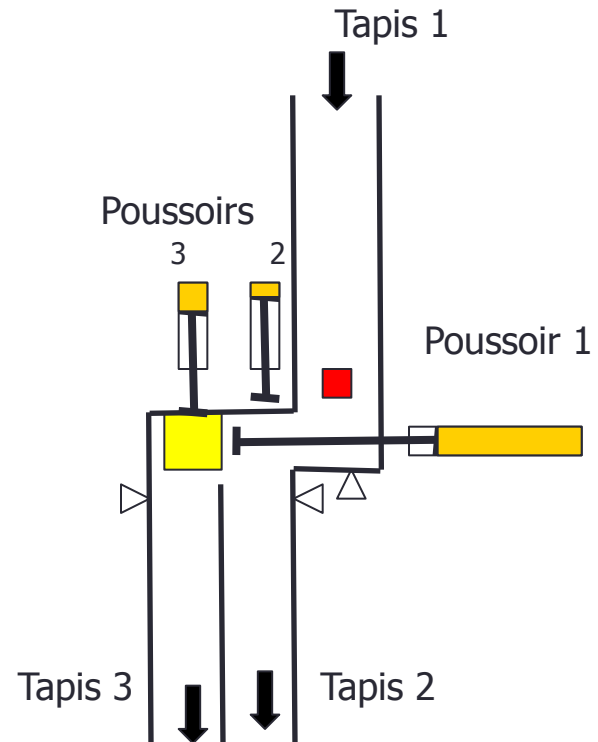
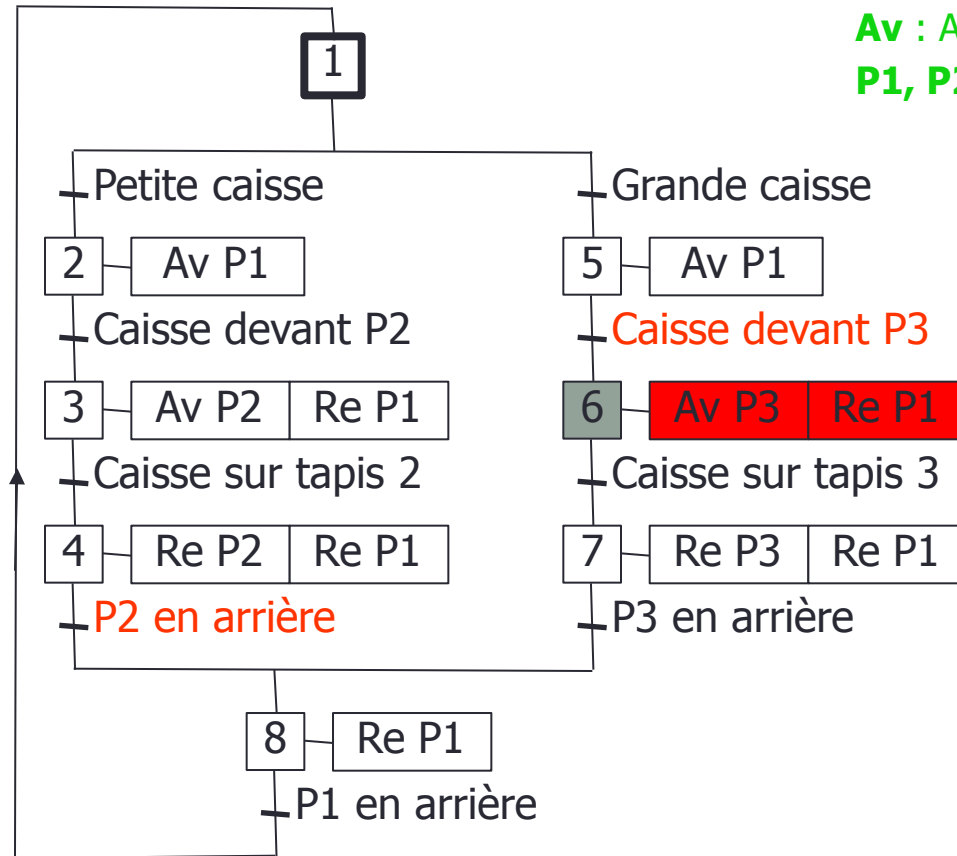
Exemple avec branchement OU (sélection de séquences)

Av : Avance Re : Recule
P1, P2, P3 : poussoirs 1, 2, 3



Exemple avec branchement OU (sélection de séquences)

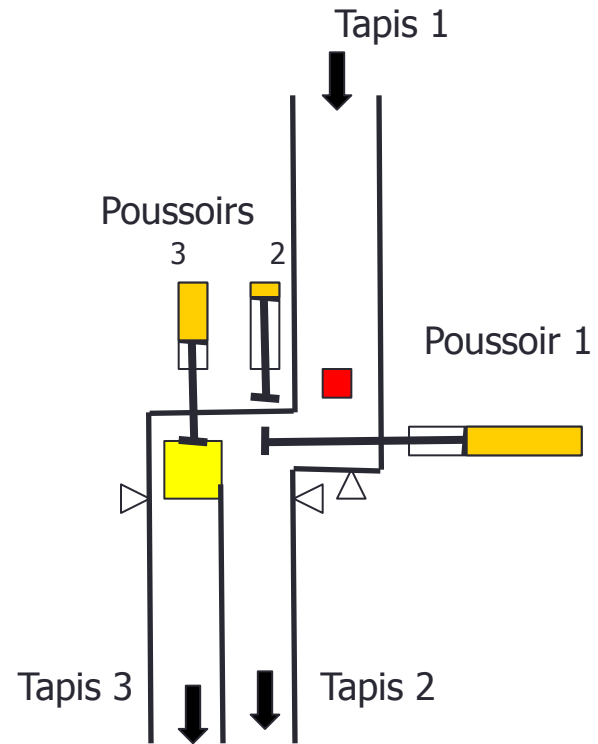
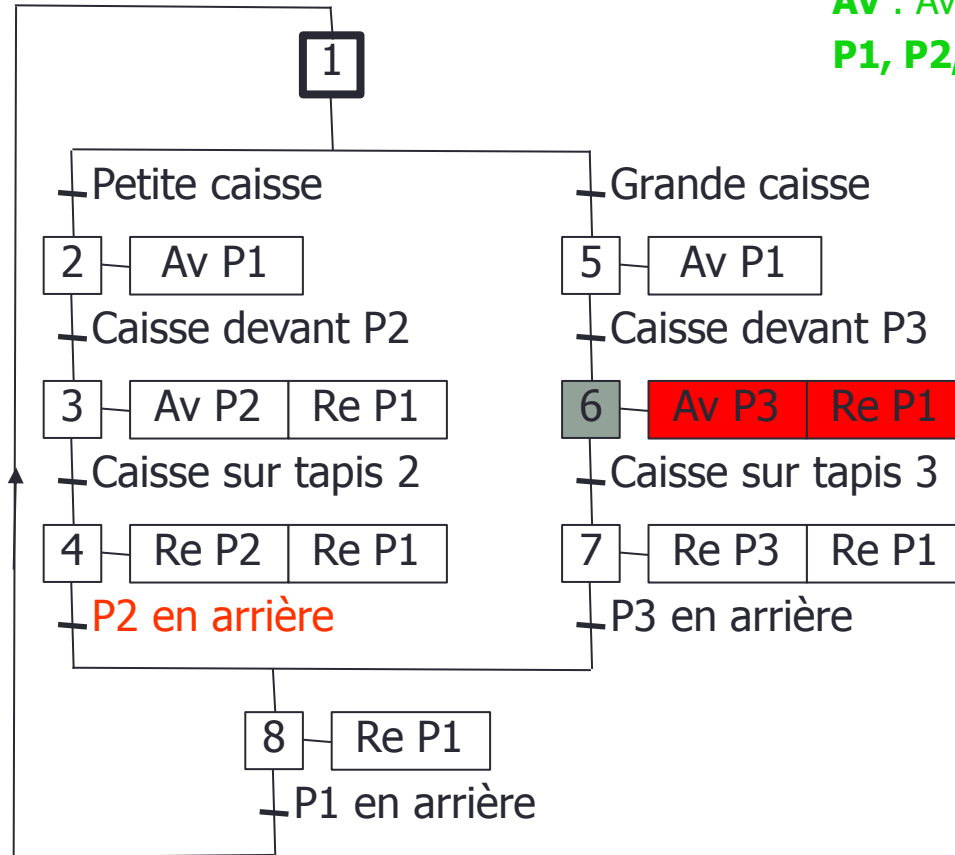
Av : Avance Re : Recule
P1, P2, P3 : poussoirs 1, 2, 3



Exemple avec branchement OU (sélection de séquences)

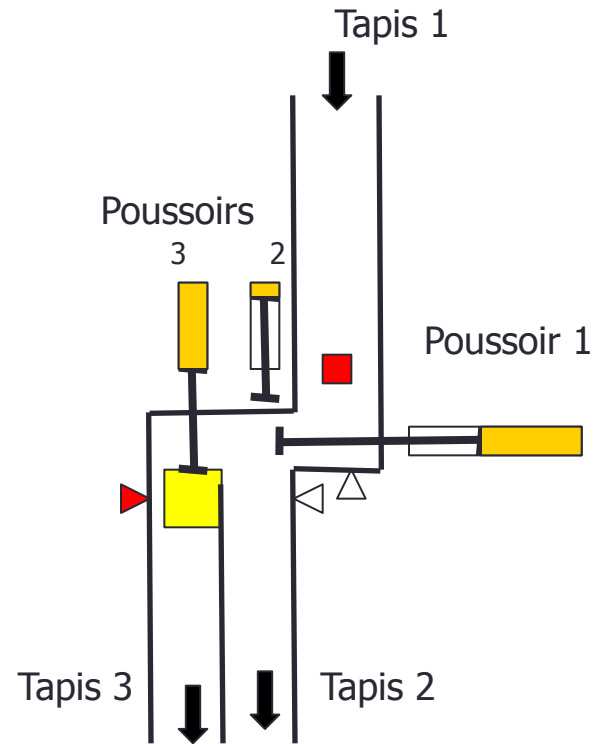
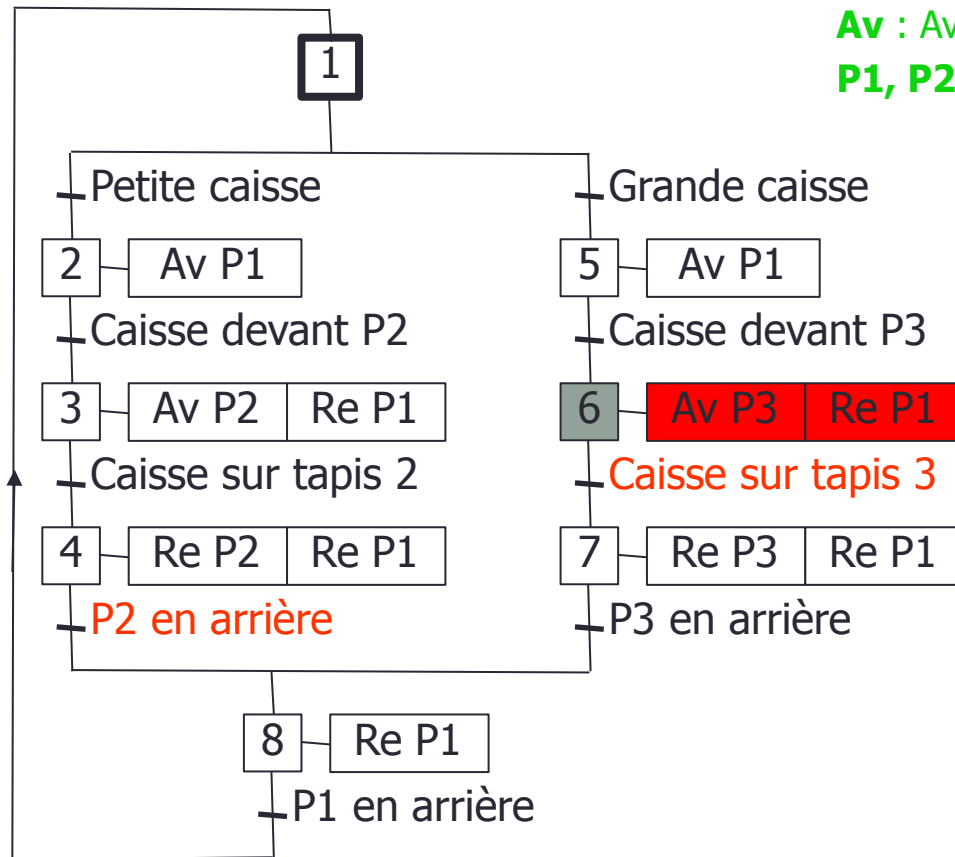
Av : Avance Re : Recule

P1, P2, P3 : poussoirs 1, 2, 3

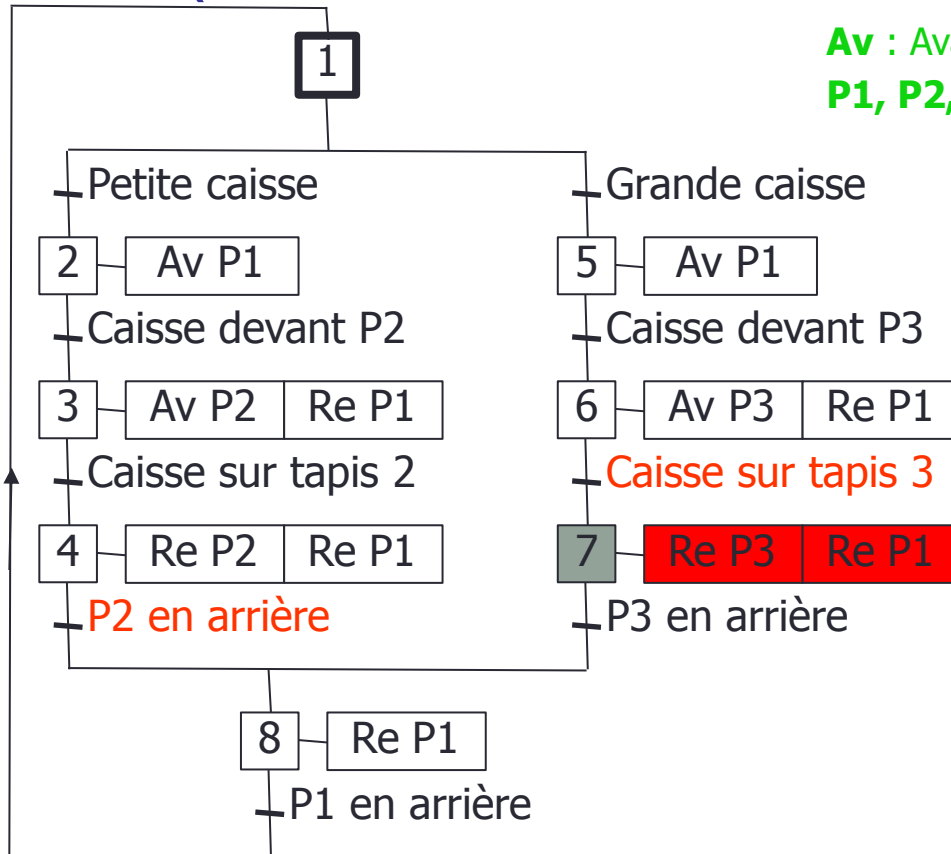


Exemple avec branchement OU (sélection de séquences)

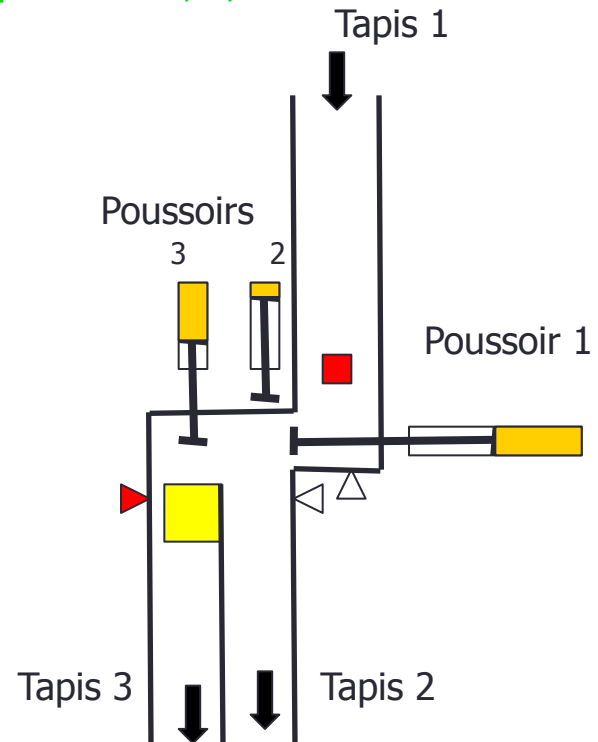
Av : Avance Re : Recule
P1, P2, P3 : poussoirs 1, 2, 3



Exemple avec branchement OU (sélection de séquences)



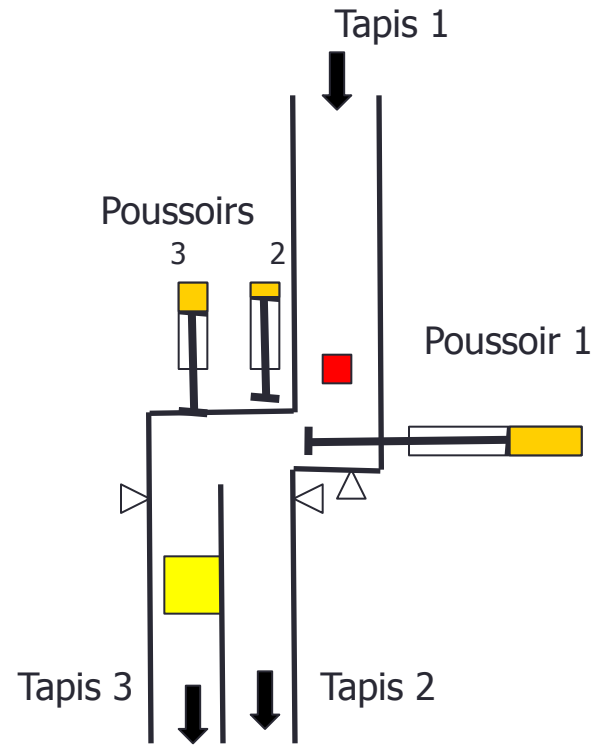
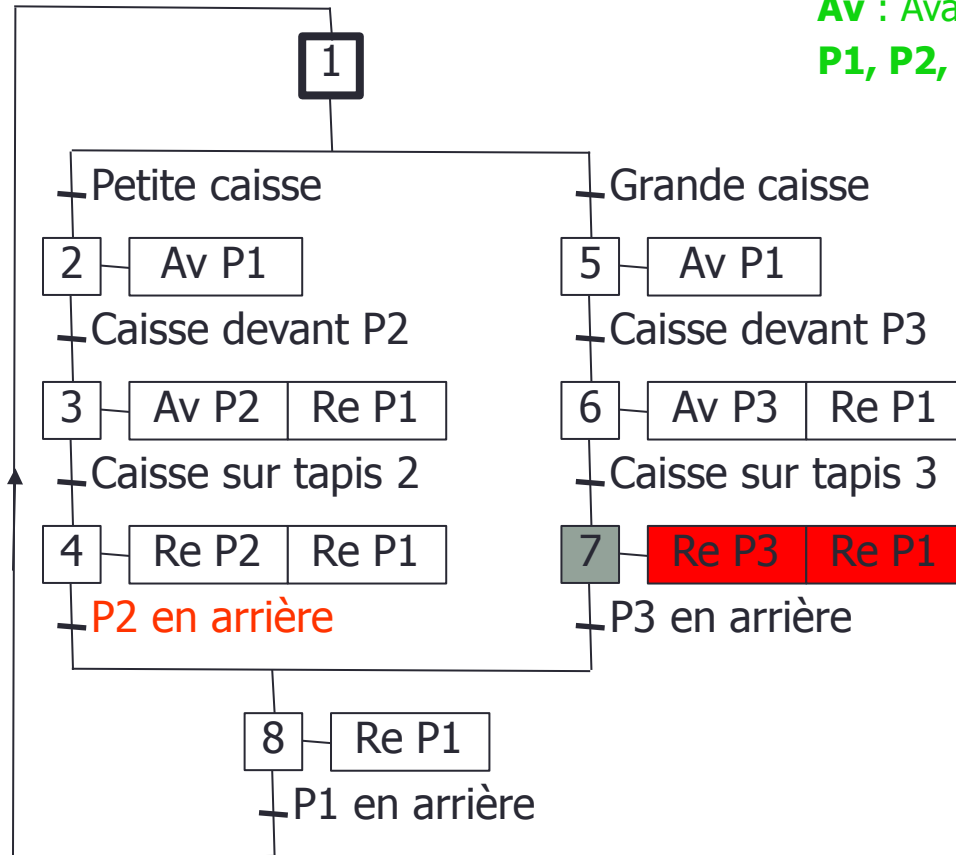
Av : Avance Re : Recule
P1, P2, P3 : poussoirs 1, 2, 3



Exemple avec branchement OU (sélection de séquences)

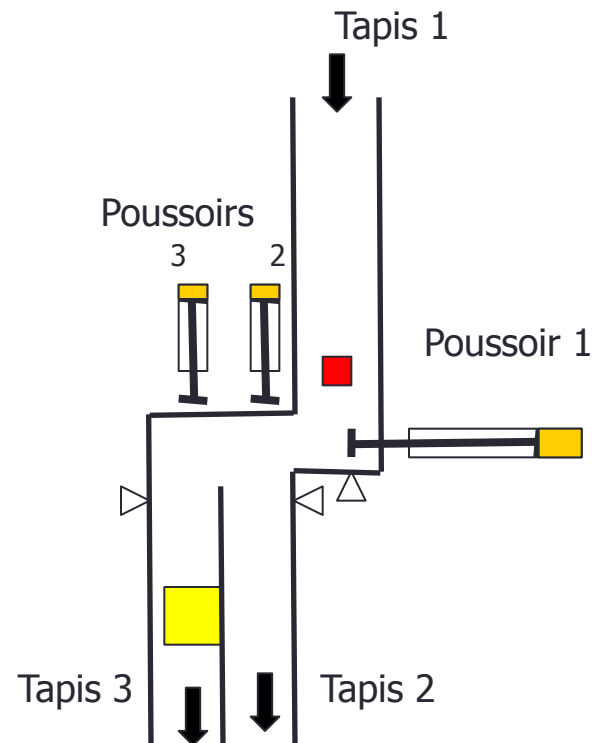
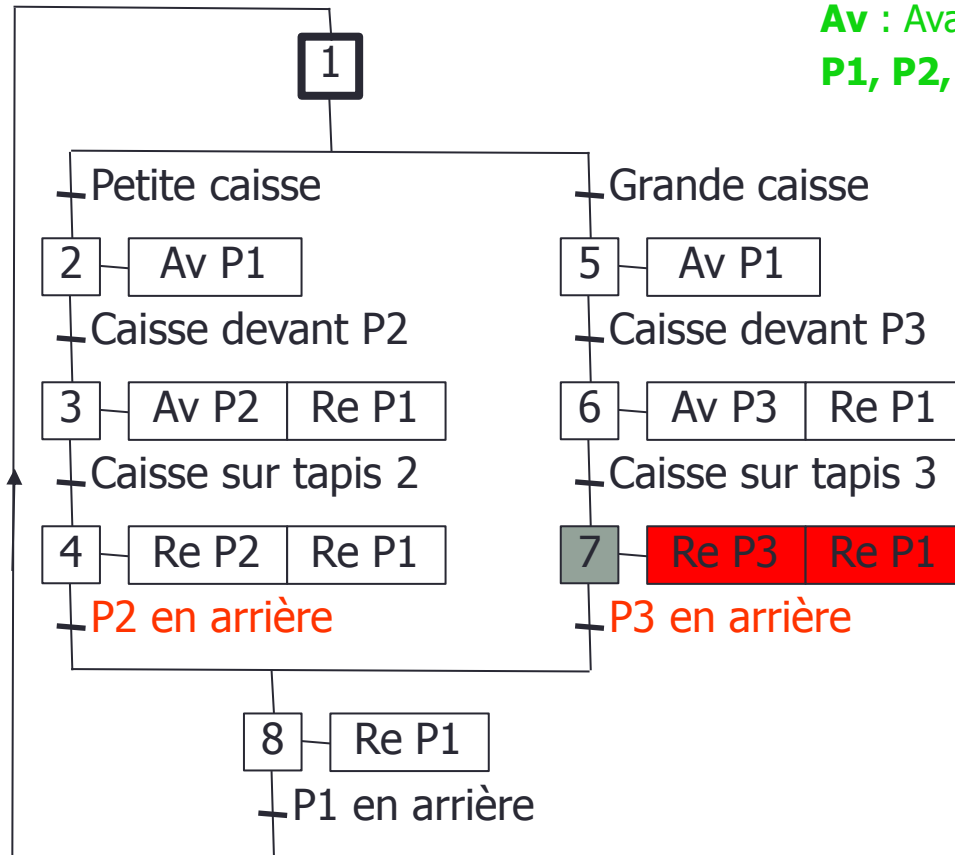
Av : Avance Re : Recule

P1, P2, P3 : poussoirs 1, 2, 3

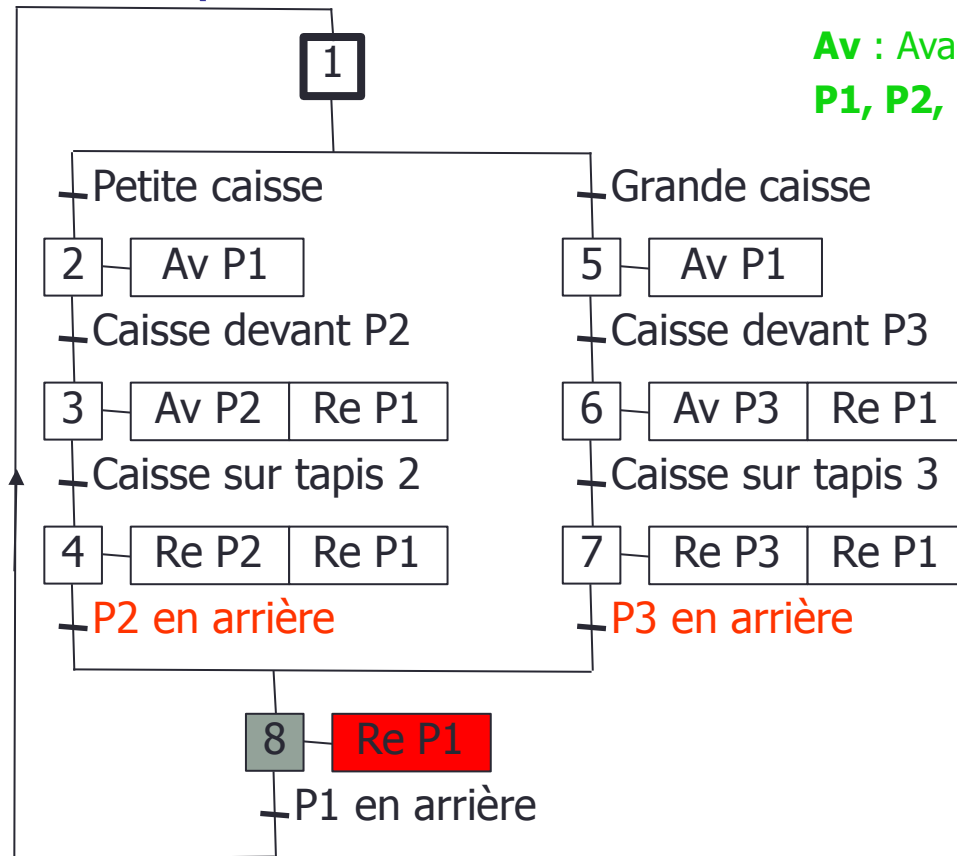


Exemple avec branchement OU (sélection de séquences)

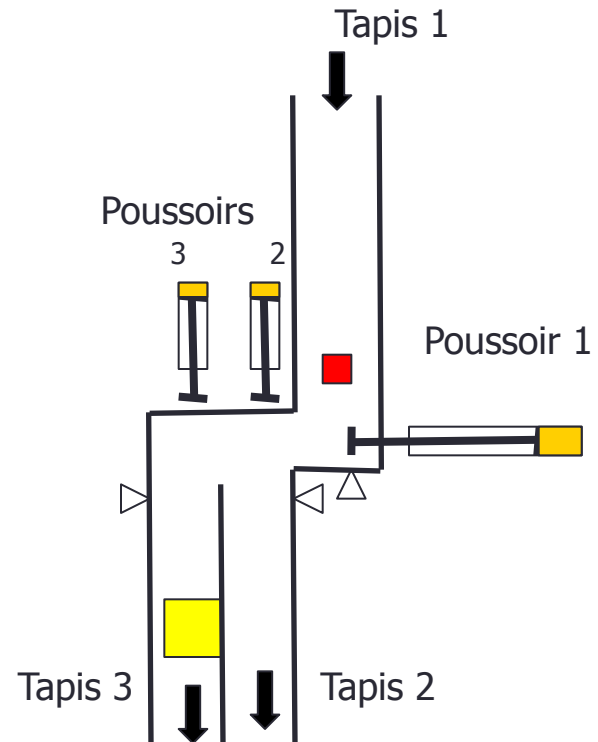
Av : Avance Re : Recule
P1, P2, P3 : poussoirs 1, 2, 3



Exemple avec branchement OU (sélection de séquences)

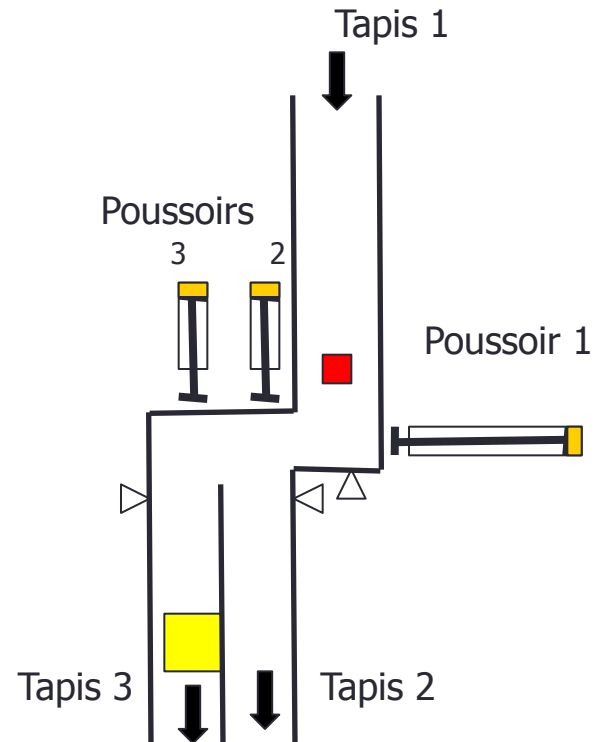
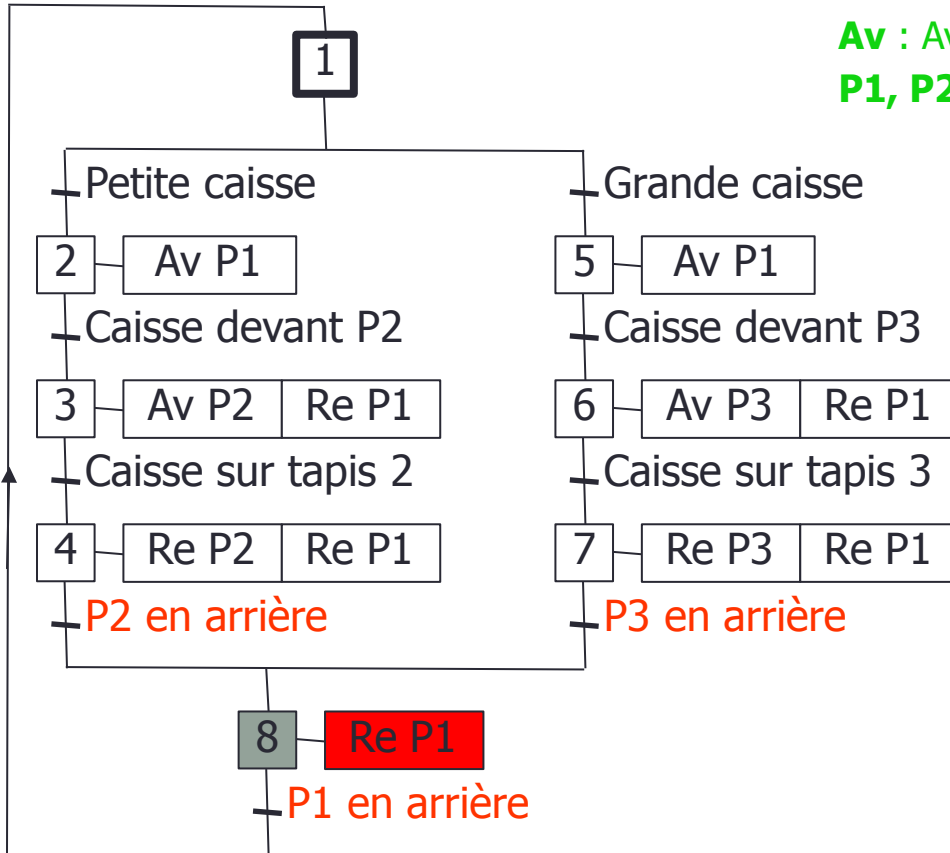


Av : Avance **Re** : Recule
P1, P2, P3 : poussoirs 1, 2, 3



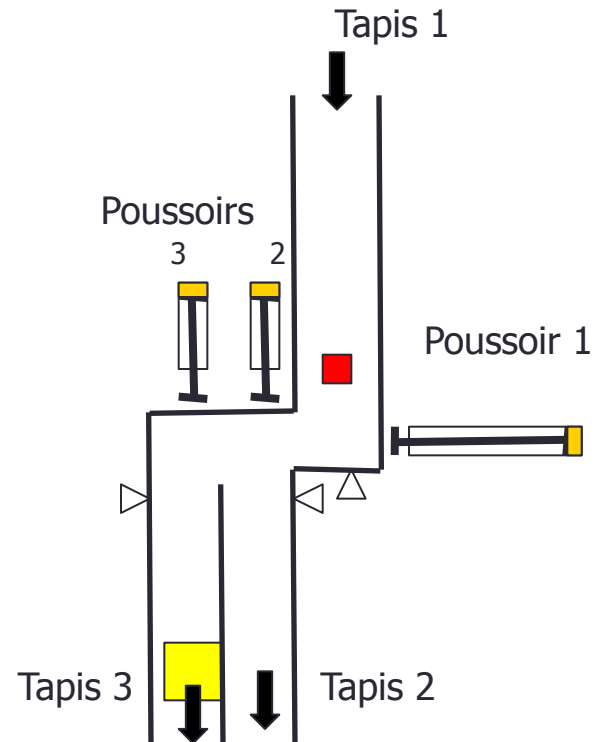
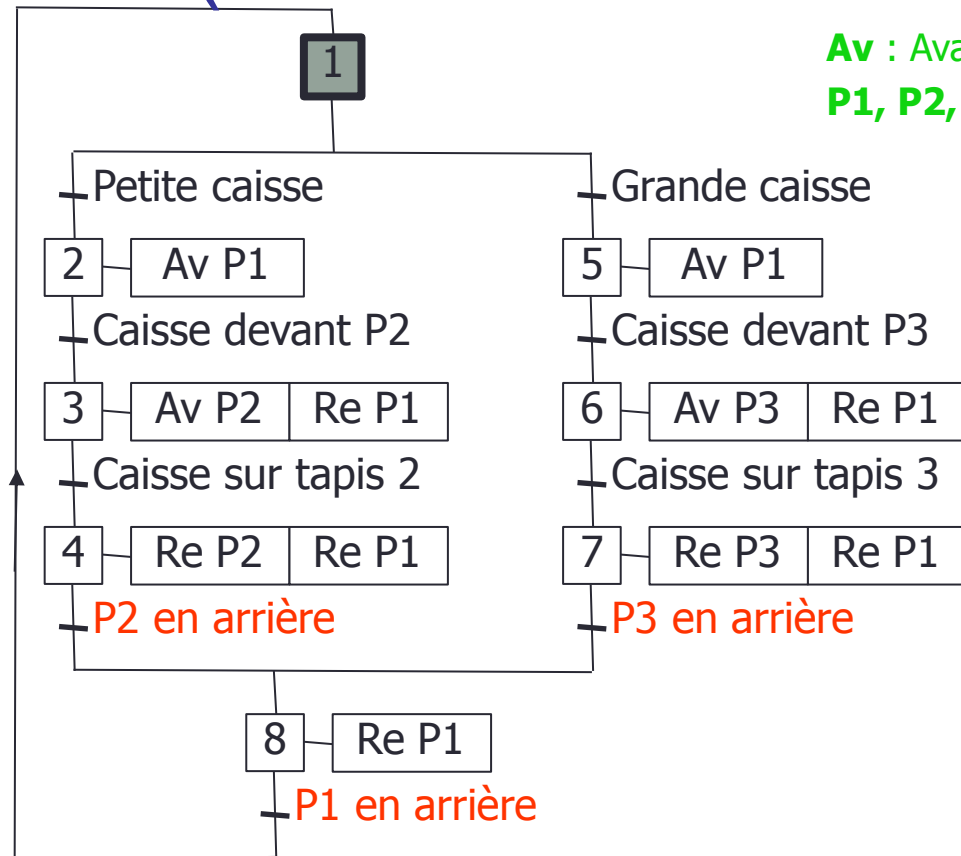
Exemple avec branchement OU (sélection de séquences)

Av : Avance **Re** : Recule
P1, P2, P3 : poussoirs 1, 2, 3

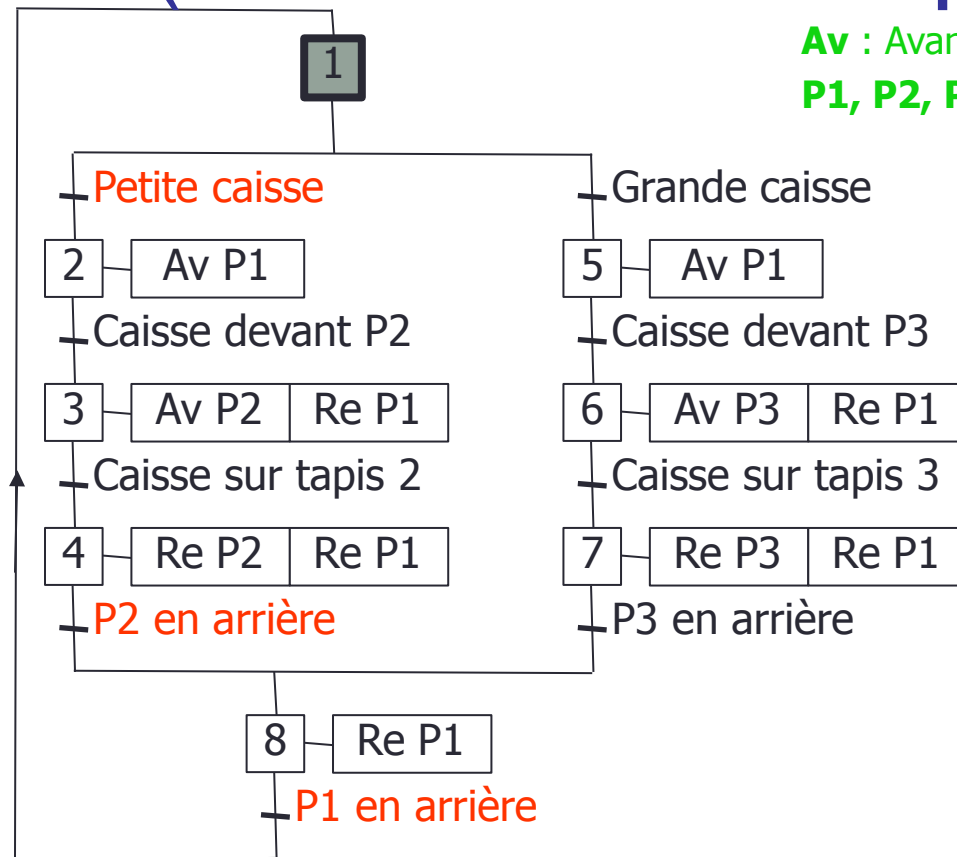


Exemple avec branchement OU (sélection de séquences)

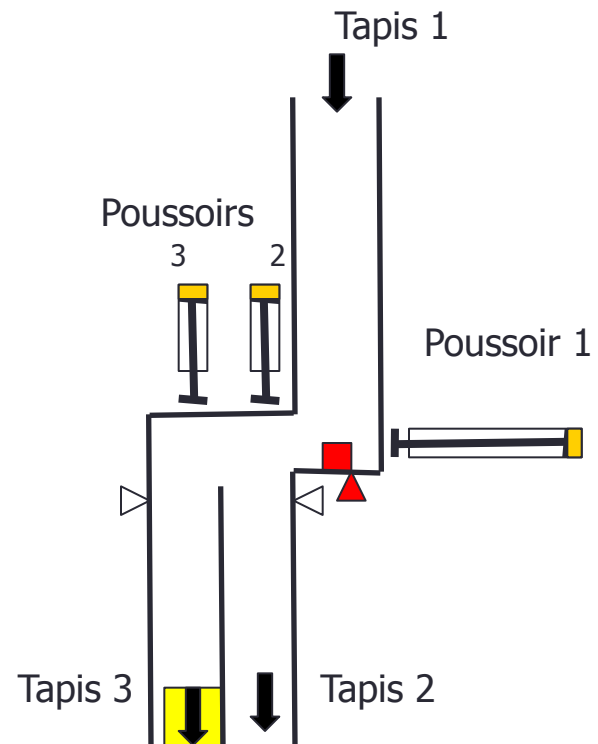
Av : Avance **Re** : Recule
P1, P2, P3 : poussoirs 1, 2, 3



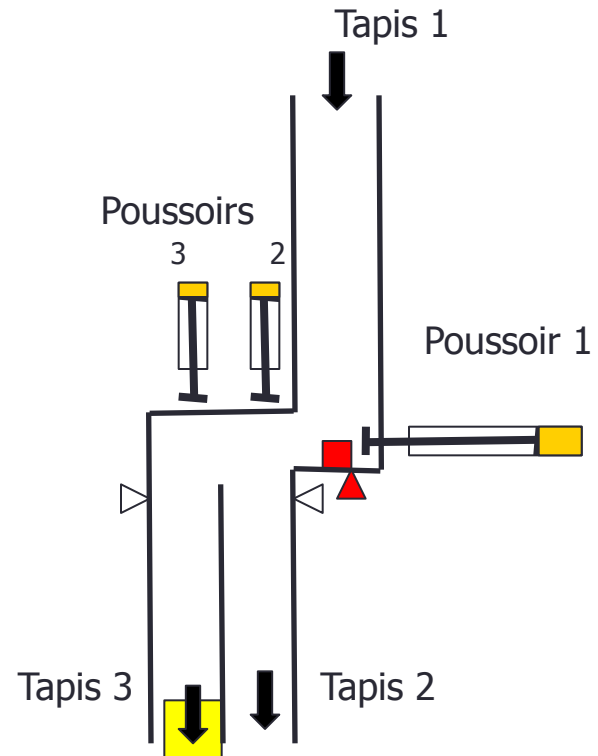
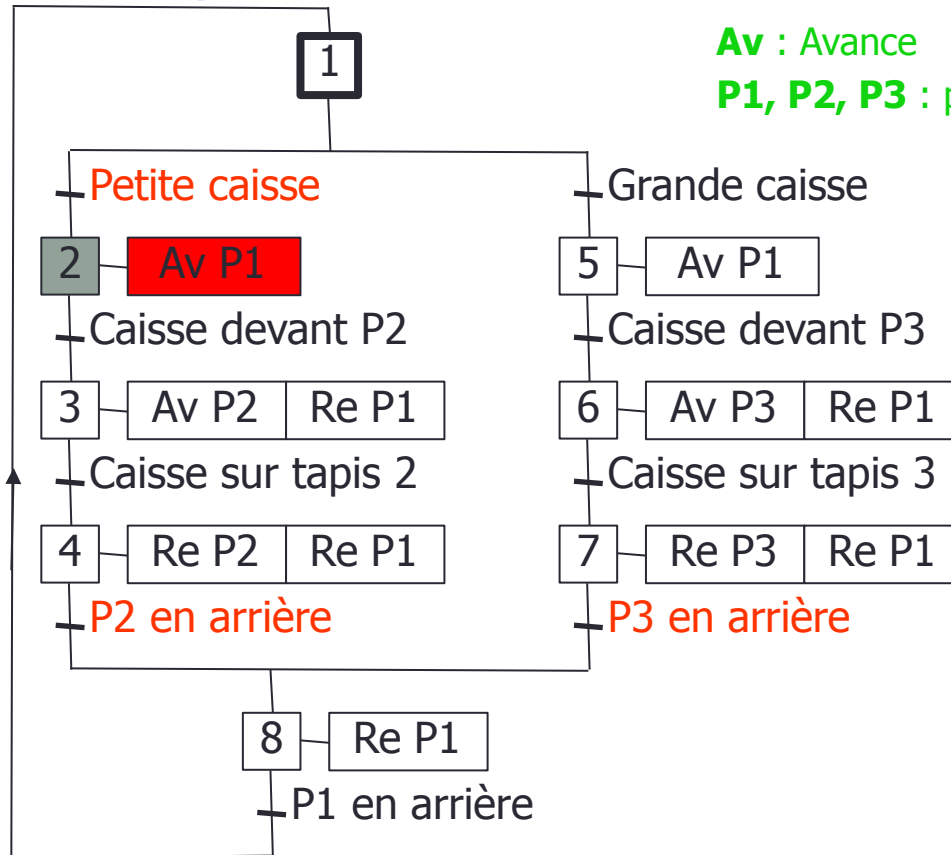
Exemple avec branchement OU (sélection de séquences)



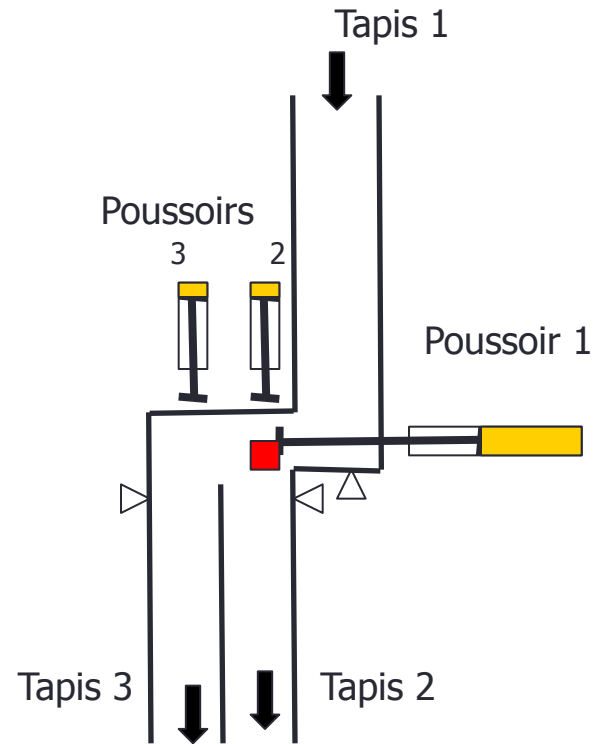
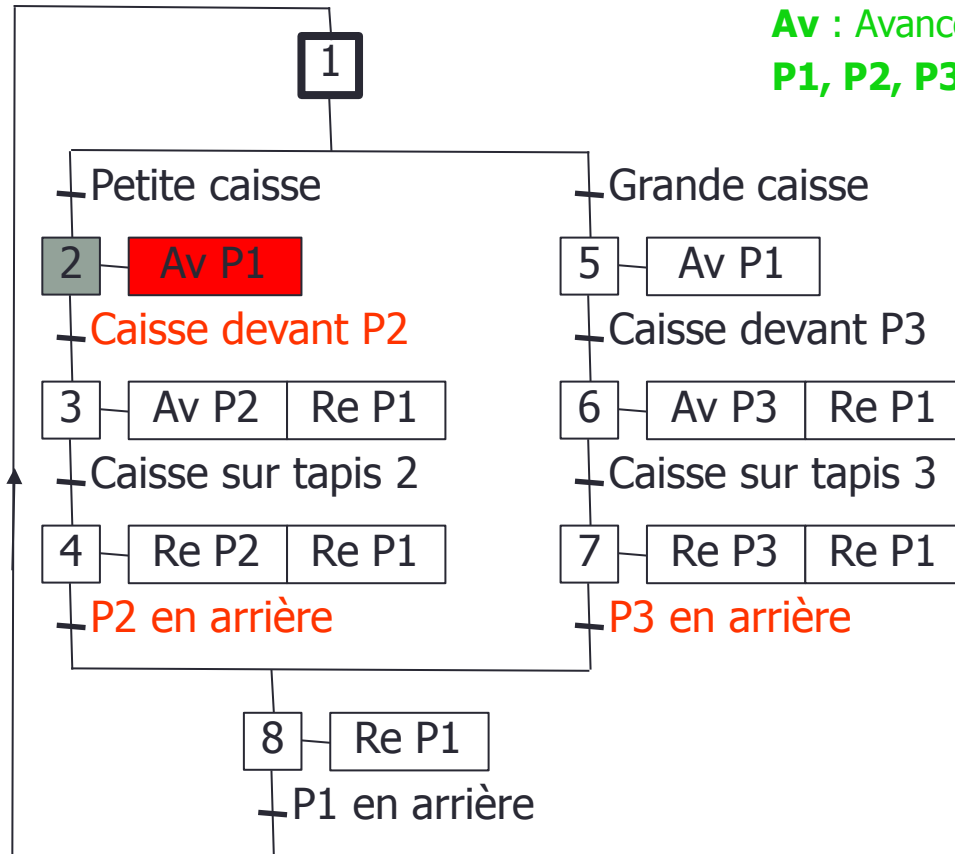
Av : Avance Re : Recule
P1, P2, P3 : poussoirs 1, 2, 3



Exemple avec branchement OU (sélection de séquences)

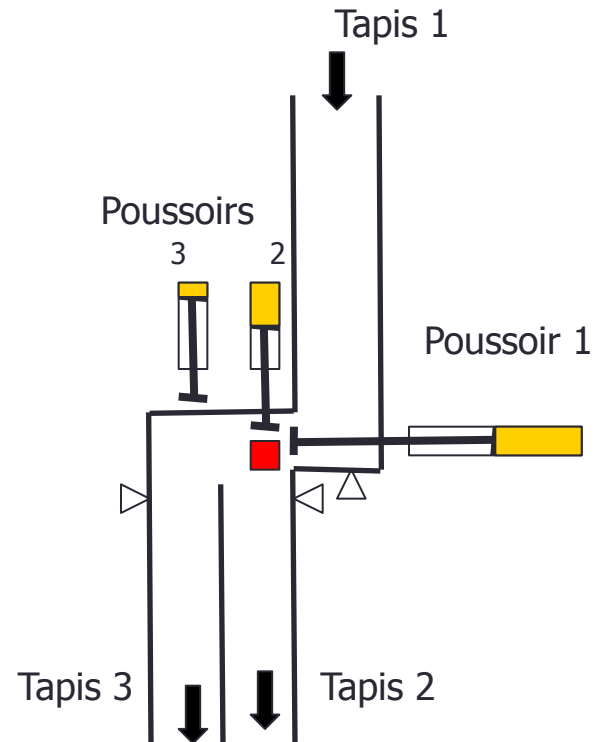
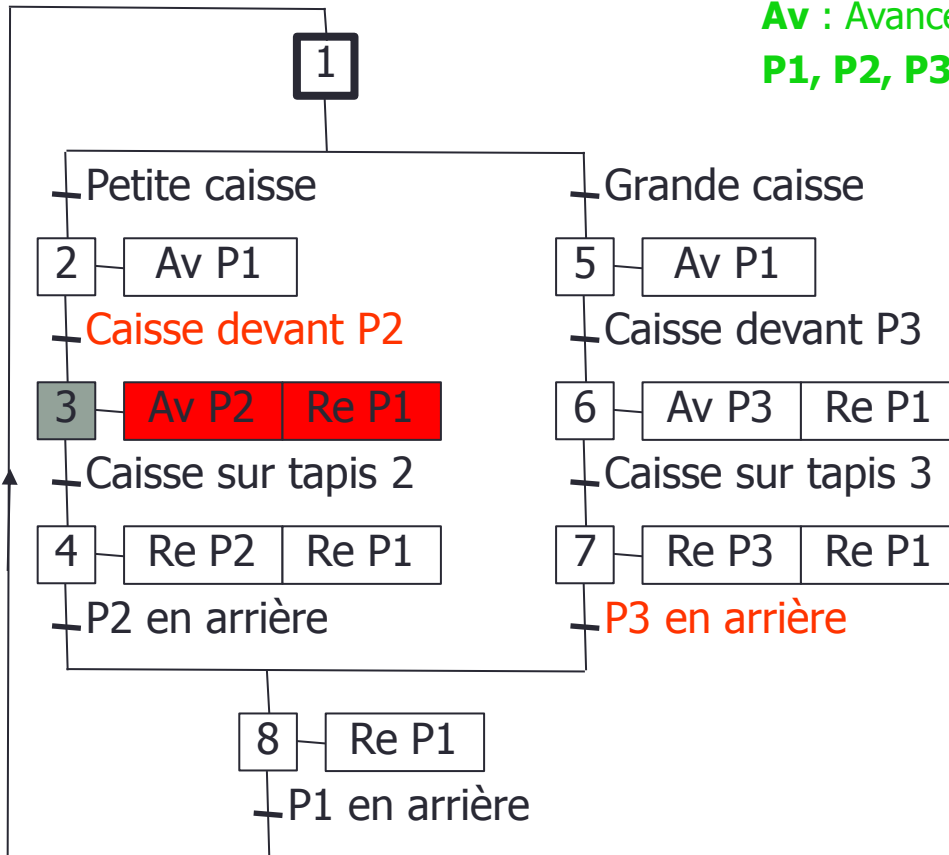


Exemple avec branchement OU (sélection de séquences)

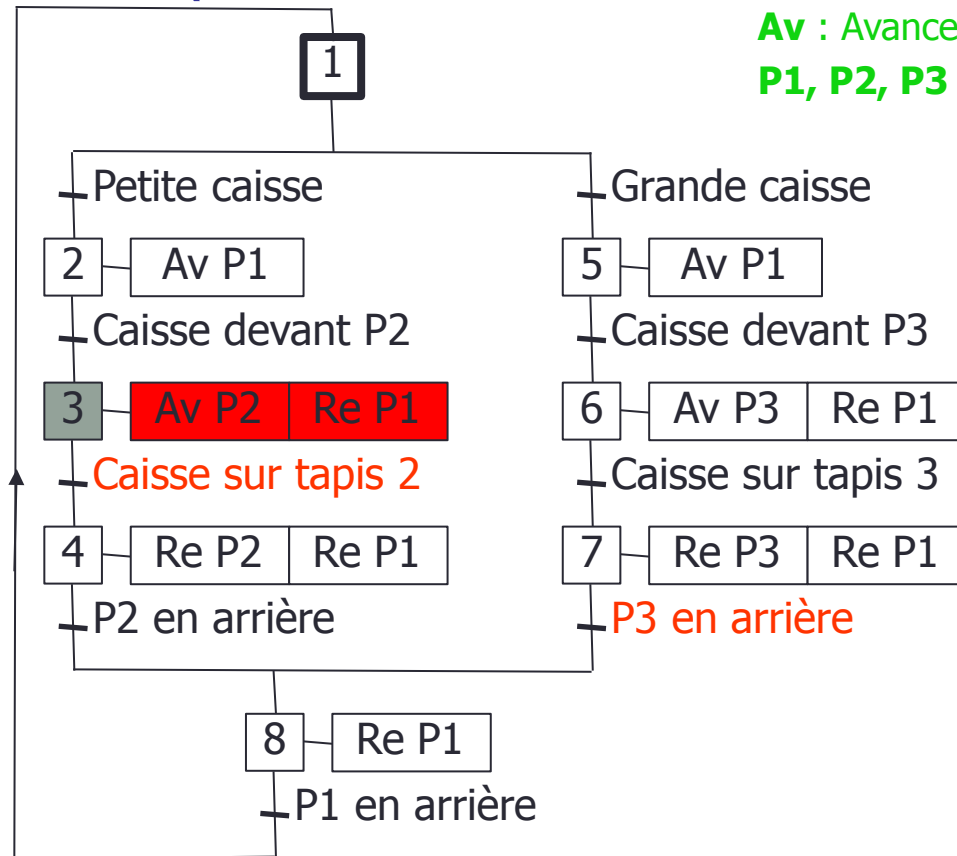


Exemple avec branchement OU (sélection de séquences)

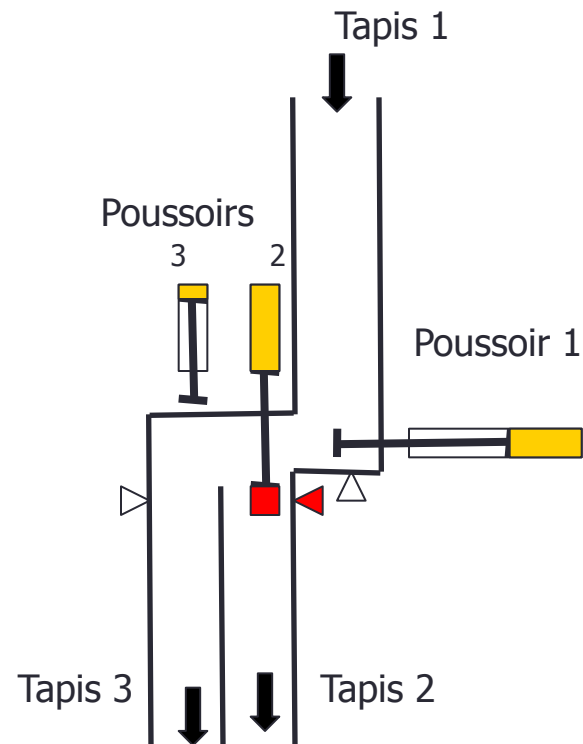
Av : Avance **Re** : Recule
P1, P2, P3 : poussoirs 1, 2, 3



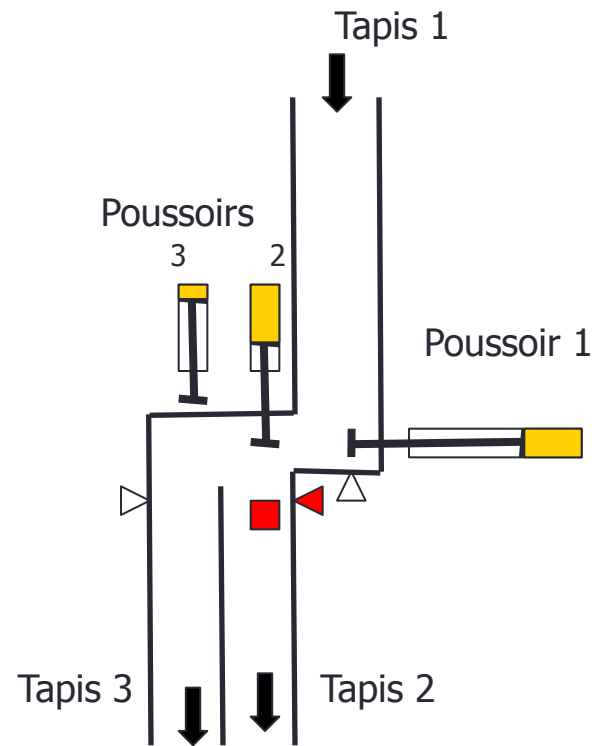
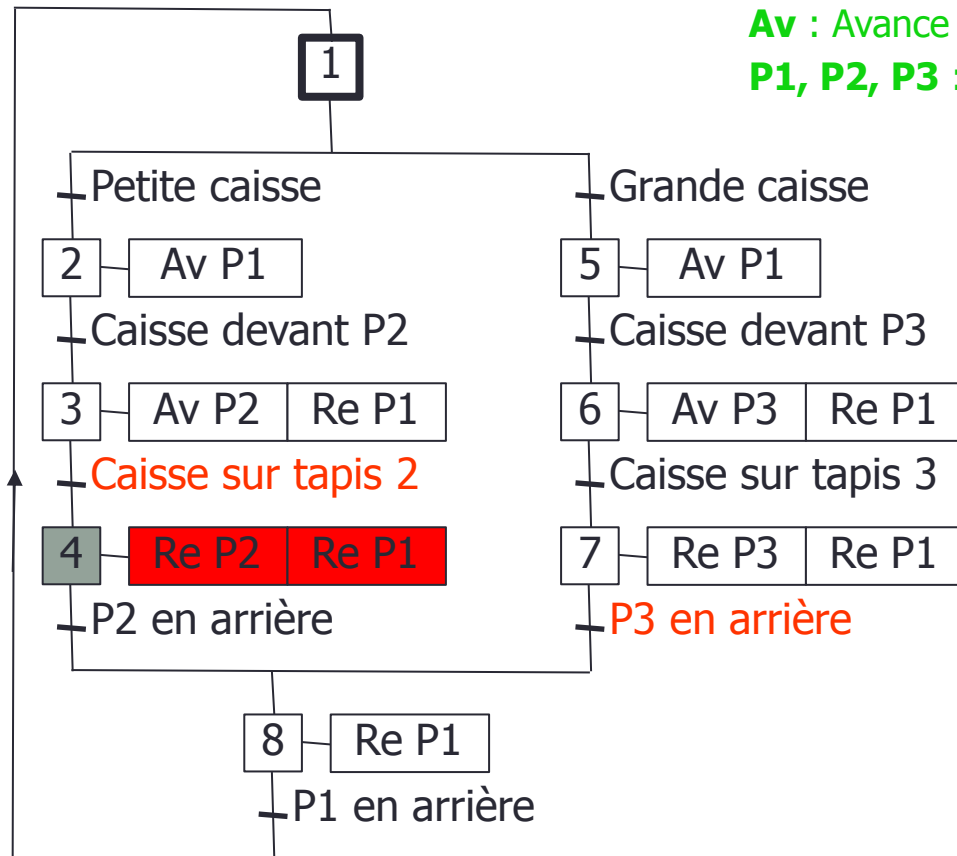
Exemple avec branchement OU (sélection de séquences)



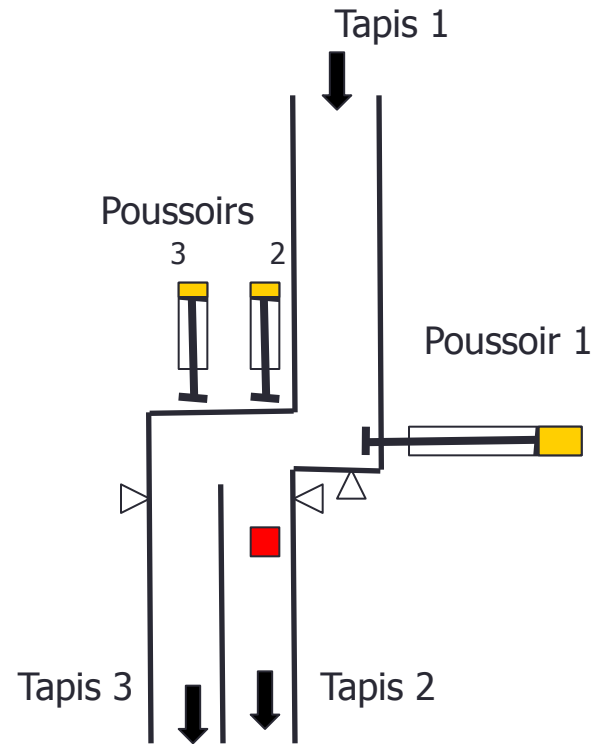
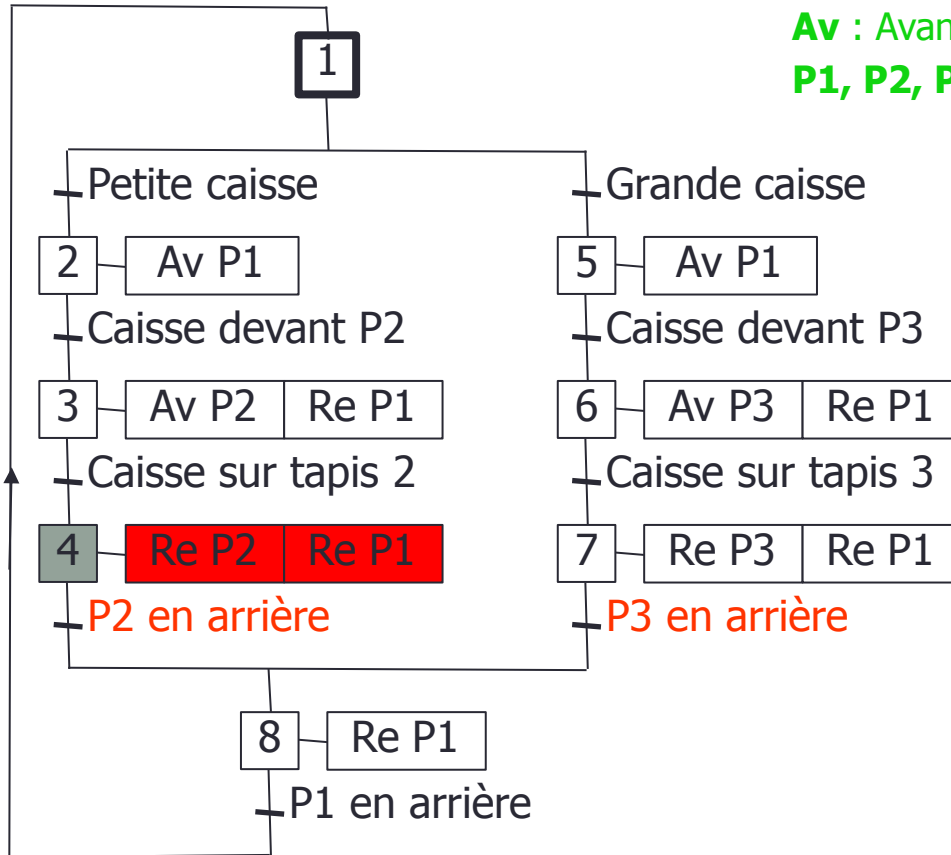
Av : Avance Re : Recule
 P1, P2, P3 : poussoirs 1, 2, 3



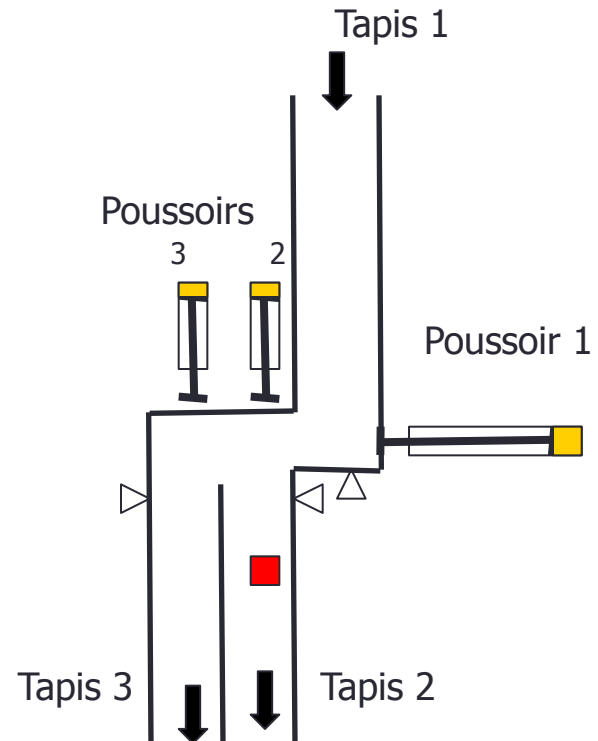
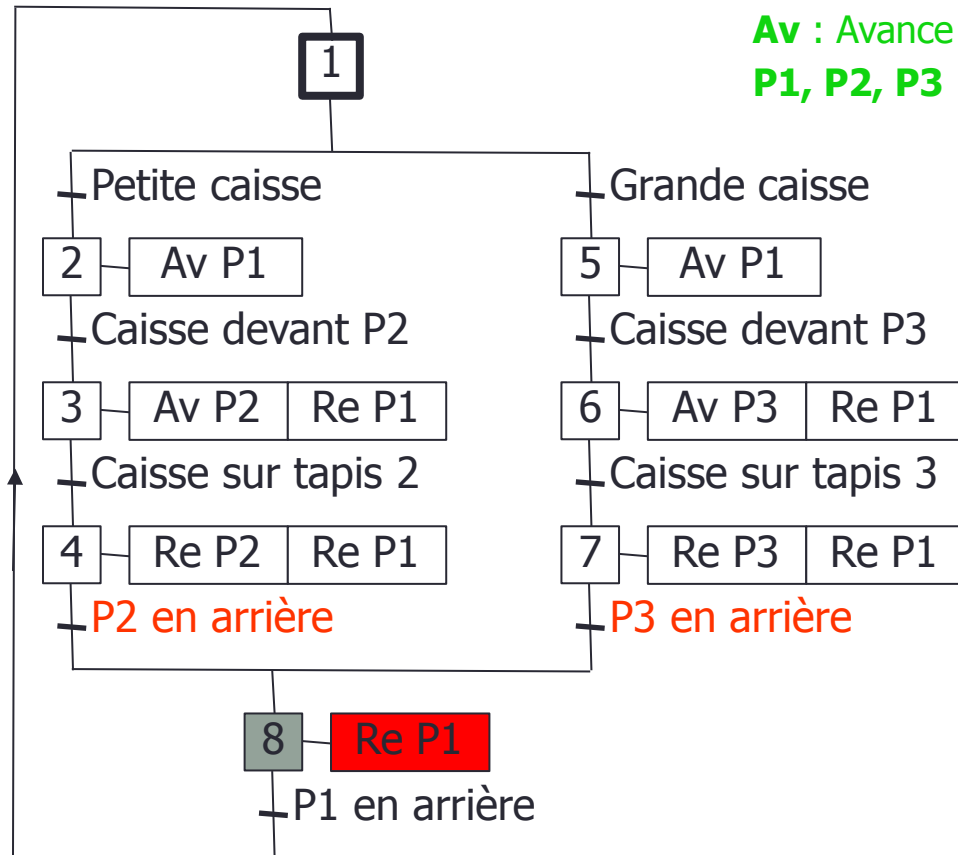
Exemple avec branchement OU (sélection de séquences)



Exemple avec branchement OU (sélection de séquences)

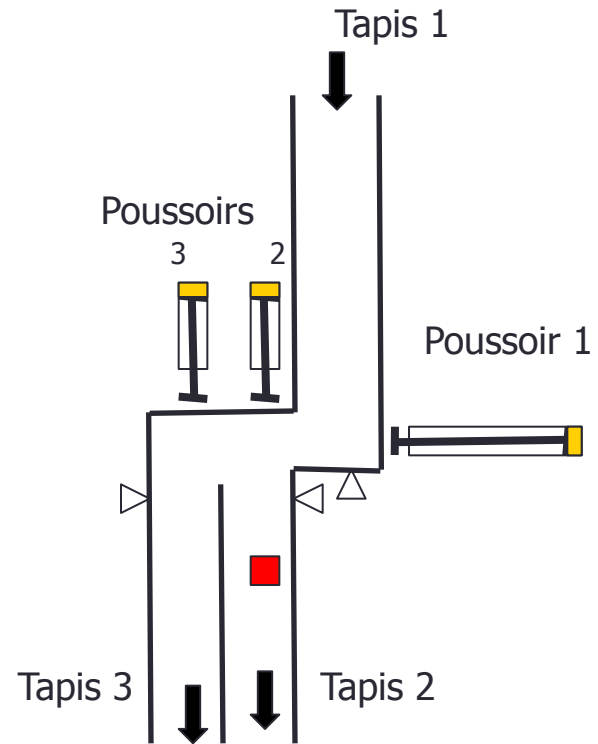
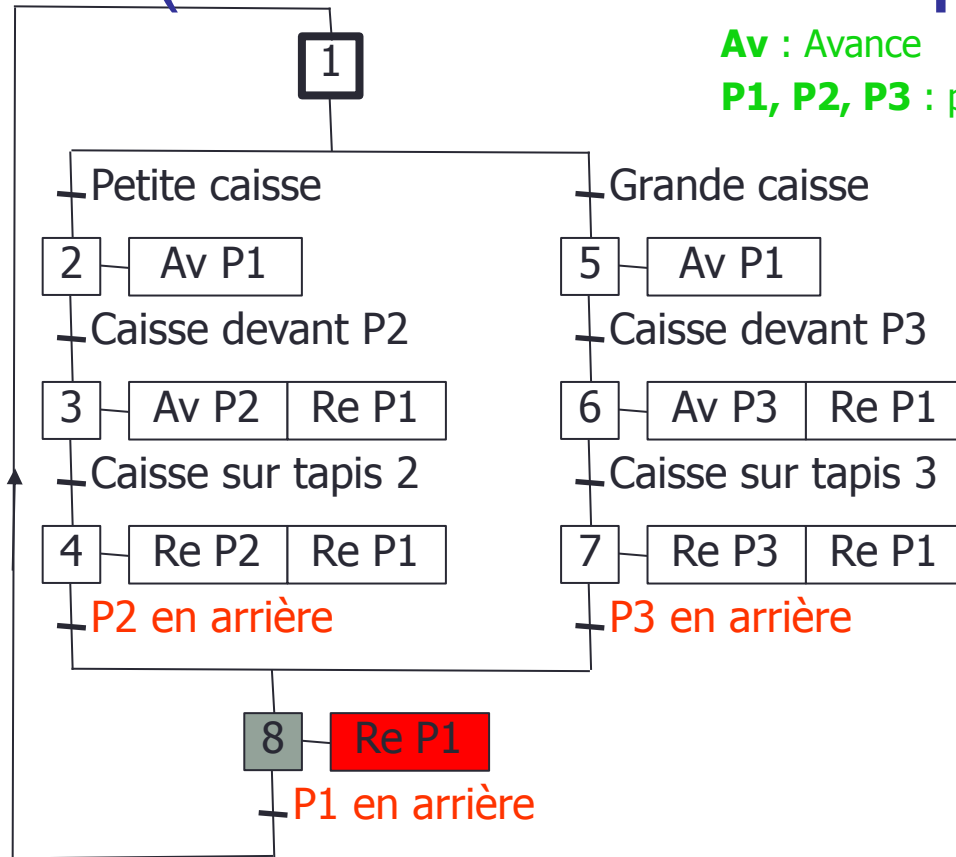


Exemple avec branchement OU (sélection de séquences)

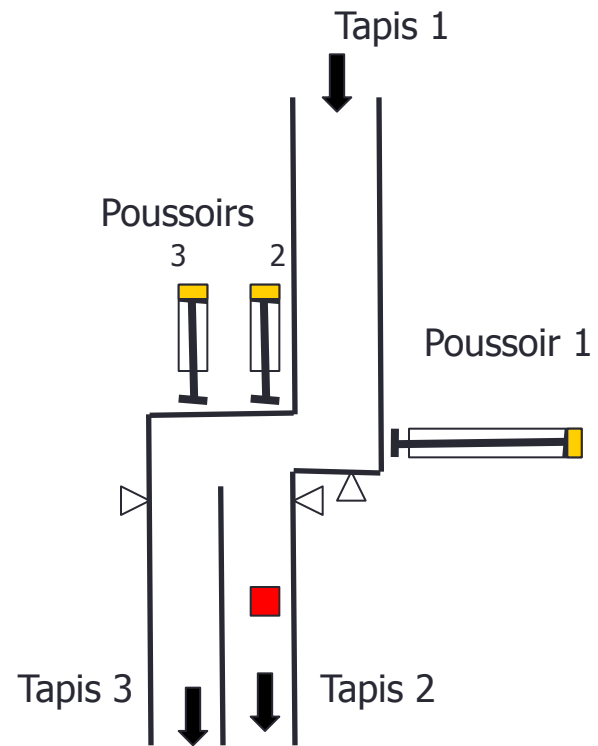
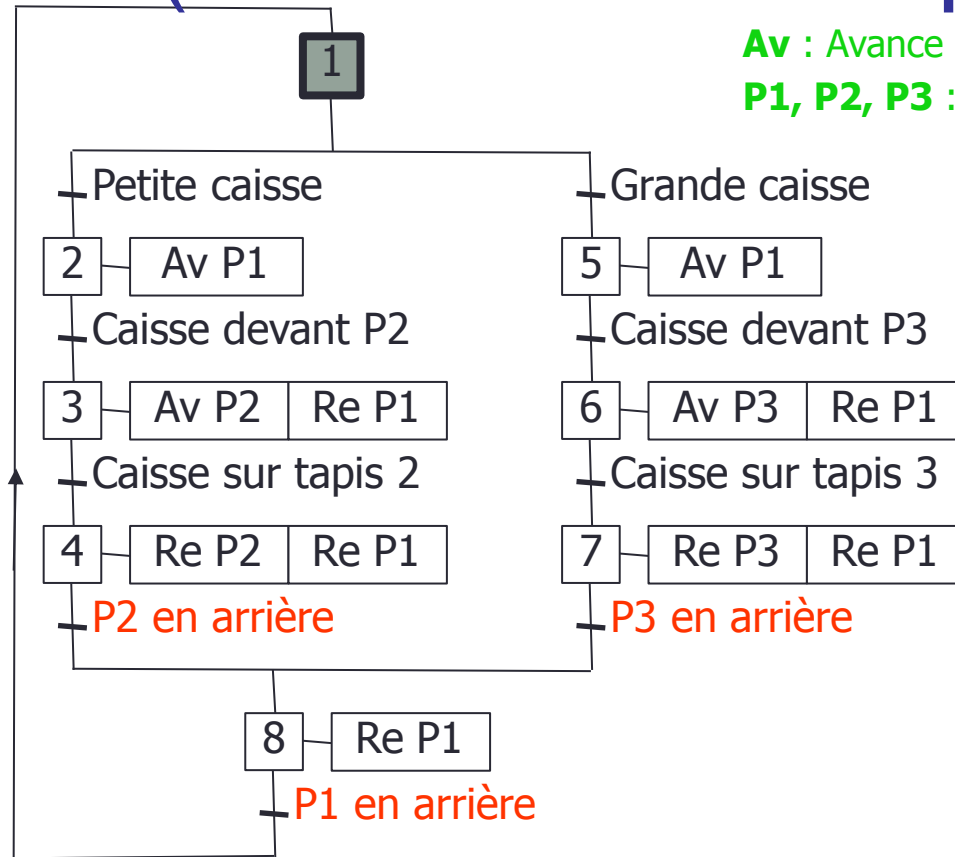


Exemple avec branchement OU (sélection de séquences)

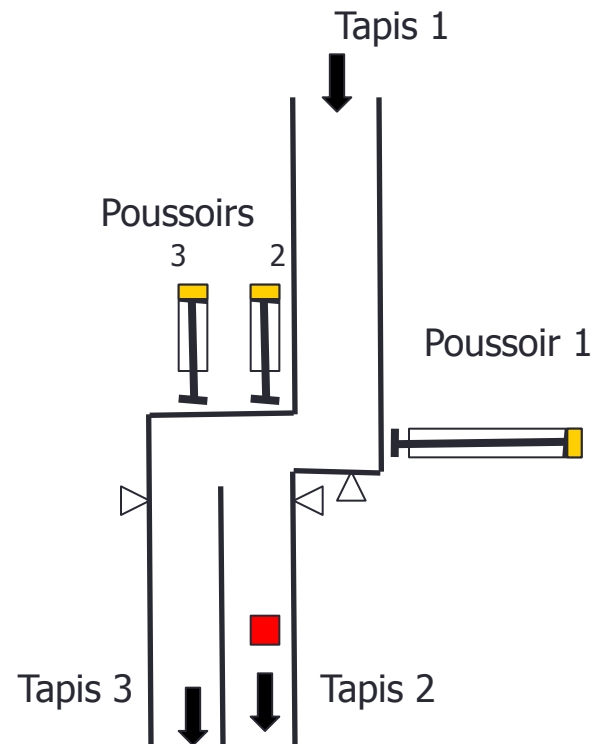
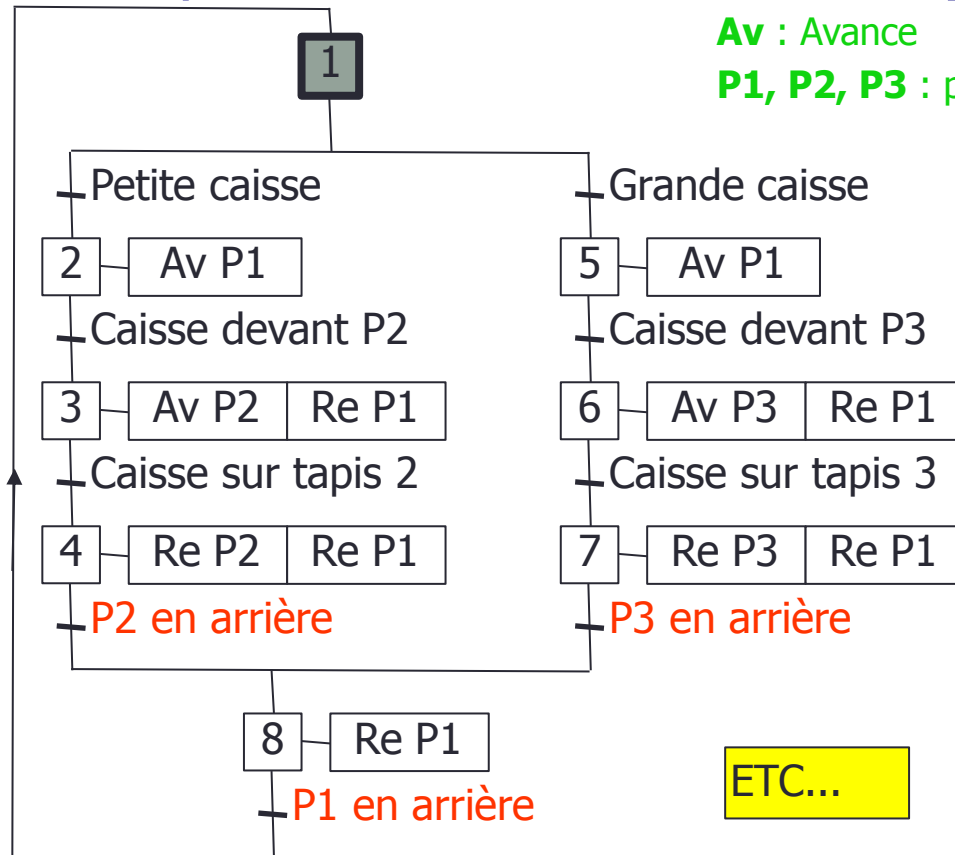
Av : Avance Re : Recule
P1, P2, P3 : poussoirs 1, 2, 3



Exemple avec branchement OU (sélection de séquences)

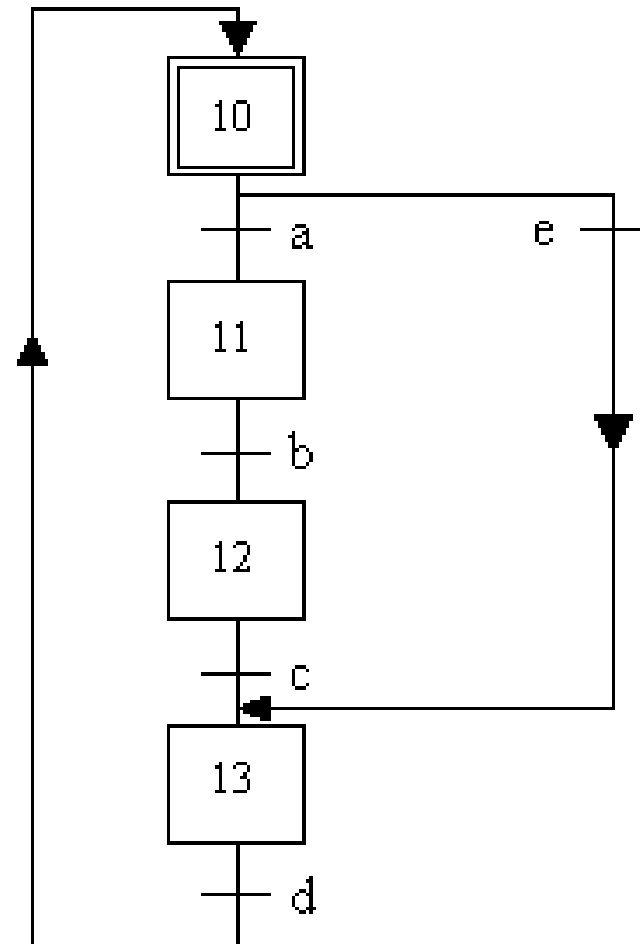


Exemple avec branchement OU (sélection de séquences)



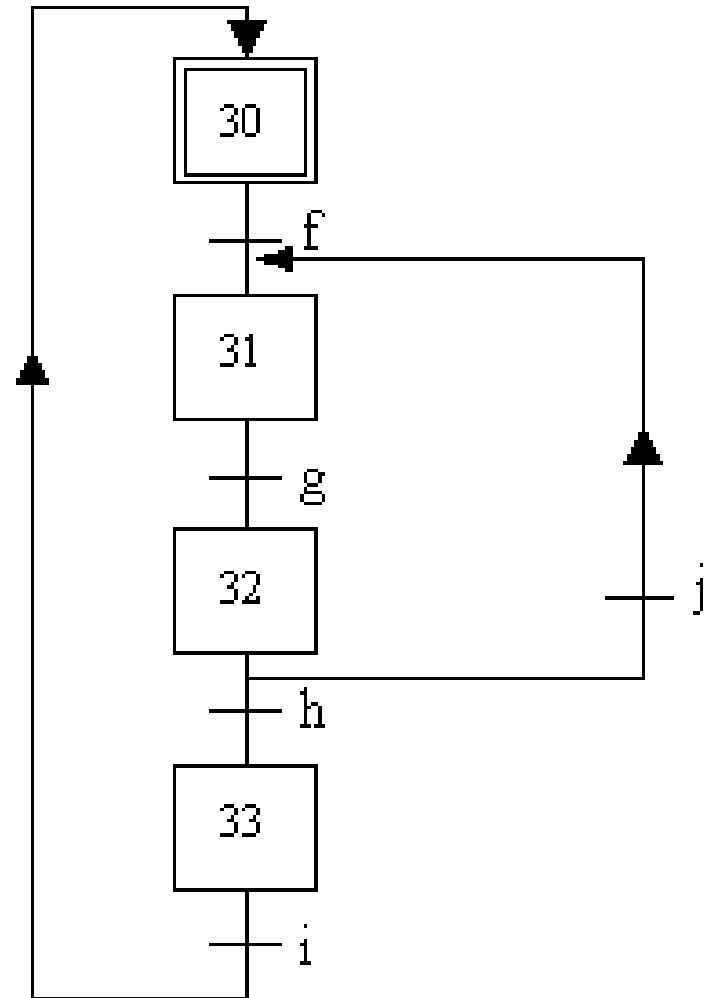
Saut en avant (saut de phase)

Le saut en avant permet de sauter une ou plusieurs étapes lorsque les actions à réaliser deviennent inutiles.



Saut en arrière (reprise de phase)

Le saut en arrière permet de reprendre une séquence lorsque les actions à réaliser sont répétitives.

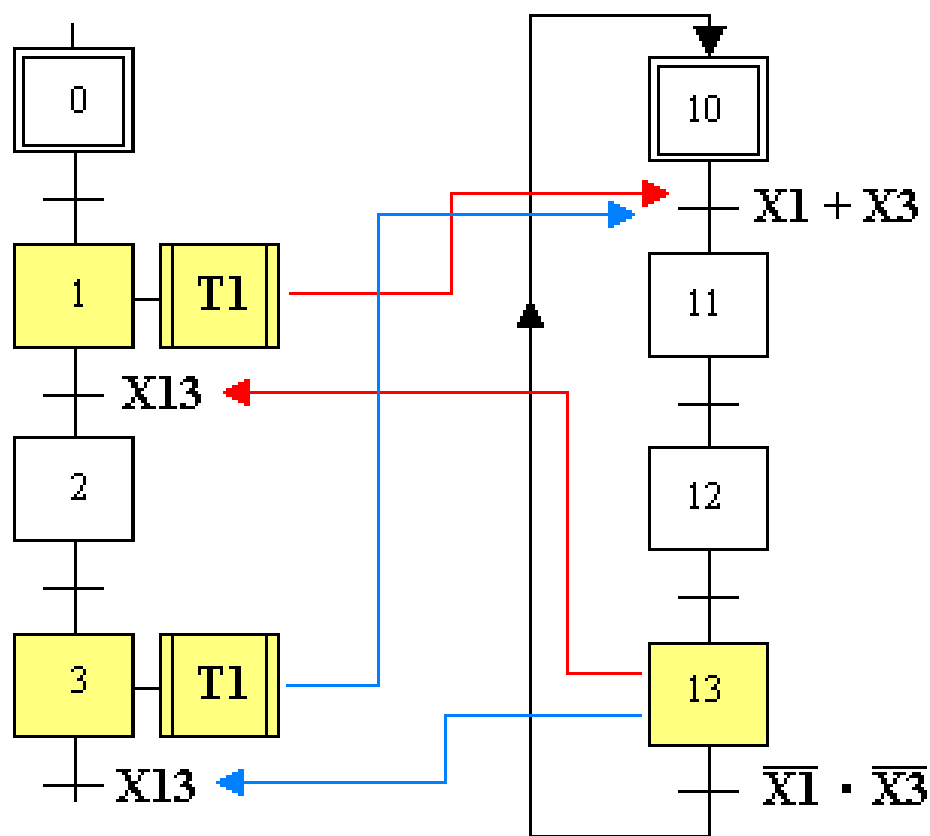


MACRO - REPRÉSENTATIONS

Sous-programme (tâche)

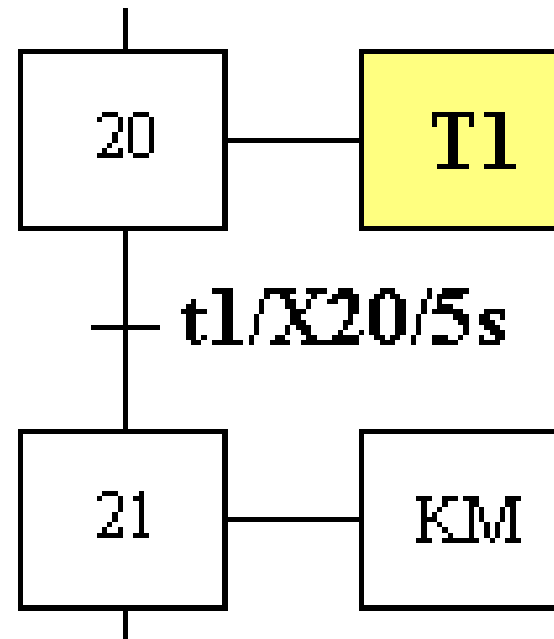
Grafcet Principal

Grafcet T1



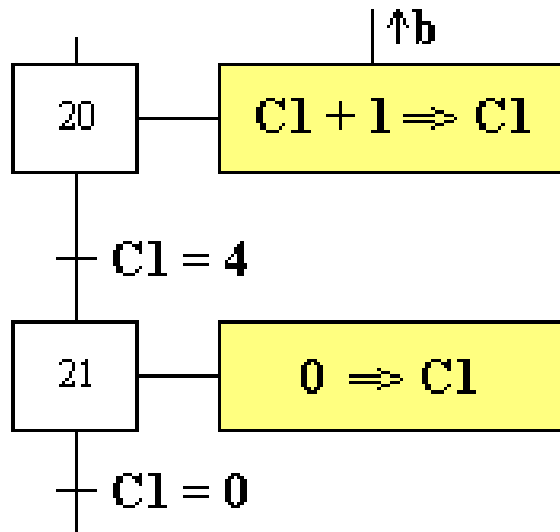
TEMPORISATIONS

La transition 20 - 21 est franchie lorsque la temporisation, démarrée à l'étape 20 est écoulée, soit au bout de 5s.

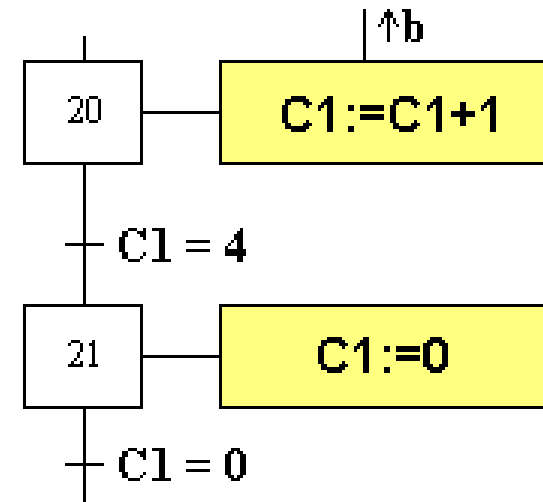


COMPTAGE

ancienne représentation:



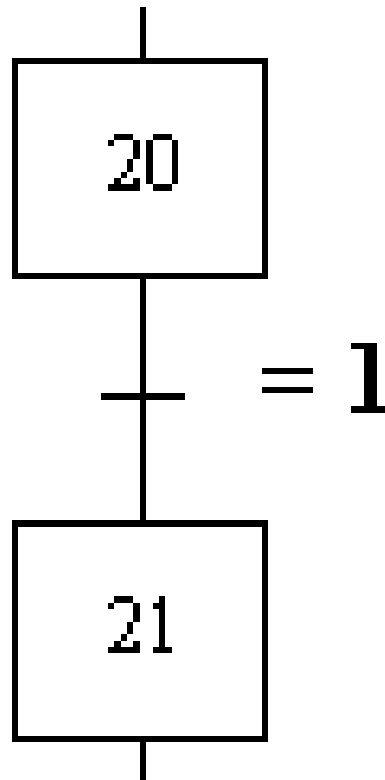
nouvelle représentation (affectation):



La transition 20 - 21 est franchie lorsque le contenu du compteur C1 est égal à 4.
Le compteur est incrémenté sur front montant du signal b.
Il est mis à zéro à l'étape 21.

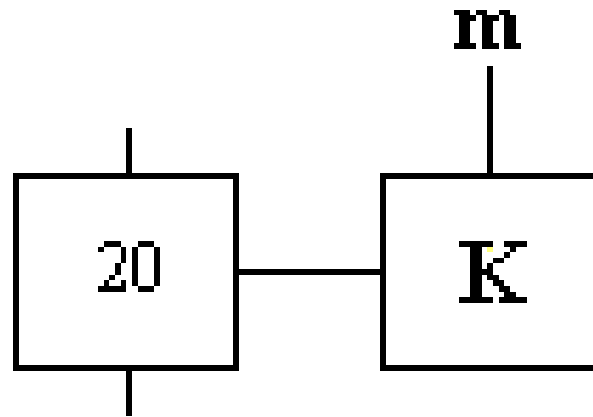
CAS PARTICULIERS

Réceptivité toujours vraie



Action conditionnelle

L'action K devient effective à l'étape 20, lorsque la condition m est vraie.
L'équation logique de K est $K = X_{20} \cdot m$

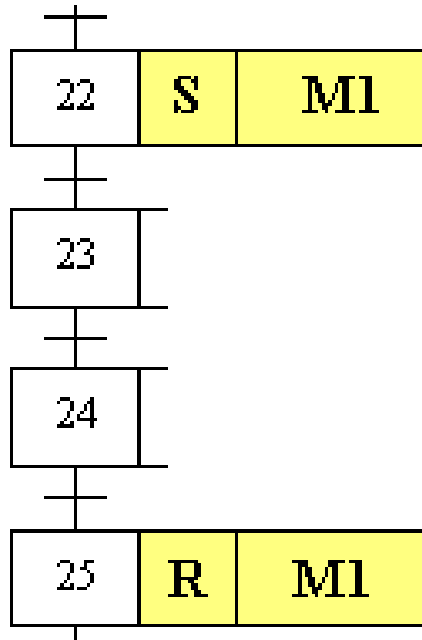


Action mémorisée

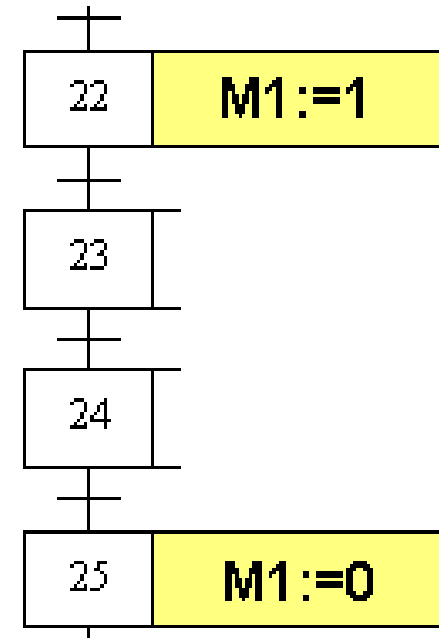
Ancienne représentation :

mise à 1 de l'action par la lettre S (set)

mise à 0 de l'action par la lettre R (reset)



Nouvelle représentation (**affectation**) :



L'action M1 est active aux étapes 22, 23 et 24.

Grafquets : compléments

- Mémorisation de passage
- Grafquet de tâche
- synchronisation horizontale
- grafquet de conduite/de tâche
- Forçage, figeage, etc ...

Mémorisation de passage

Gestion des "si condition alors"

Mémorisation de la condition (événement)

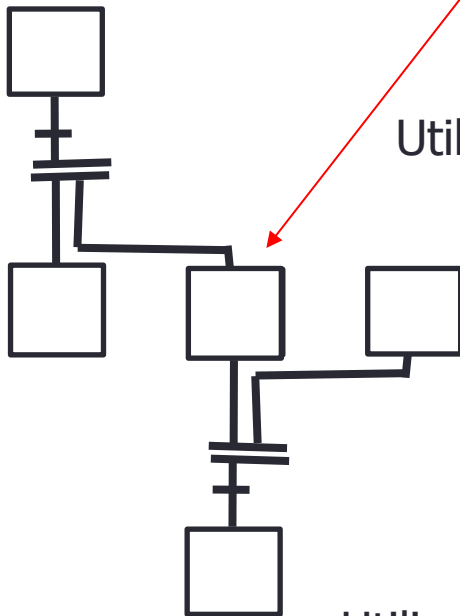


Utilisation de séquences parallèles (divergence en ET)

Utilisation de la condition (événement)



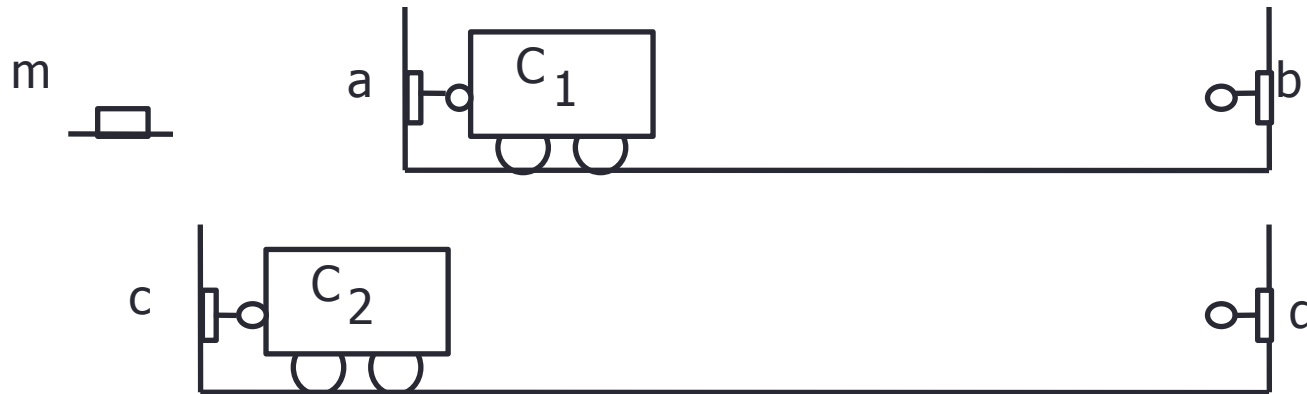
Utilisation de séquences parallèles (convergence en ET)



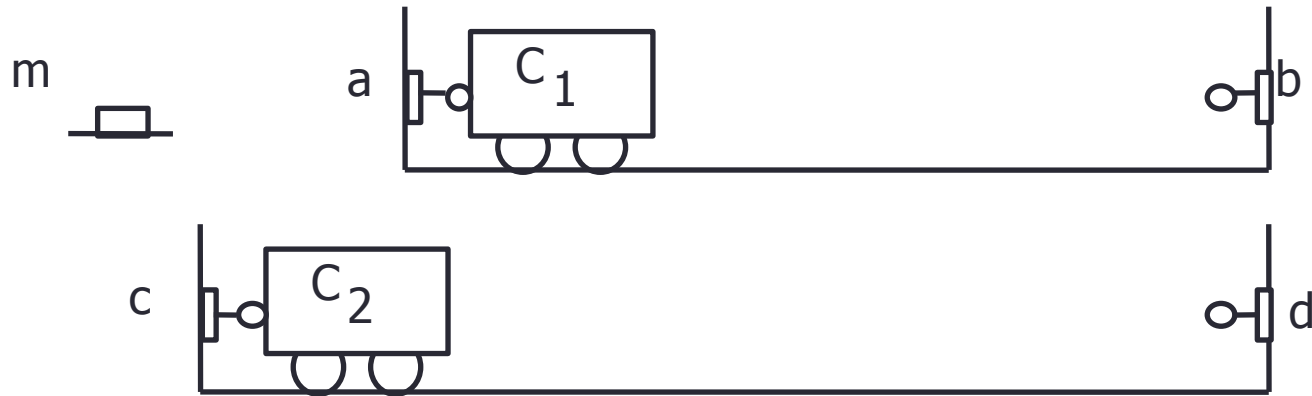
Mémorisation de passage

- **Exemple : Déchargement de deux wagonnets**

2 chariots doivent se déplacer suivant le cycle suivant :
après appui sur un bouton poussoir m les deux chariots démarrent ensemble, les chariots C1 et C2 font un aller-retour (aba) (cdc) : C1 ne peut revenir que si C2 a déjà fait un aller.



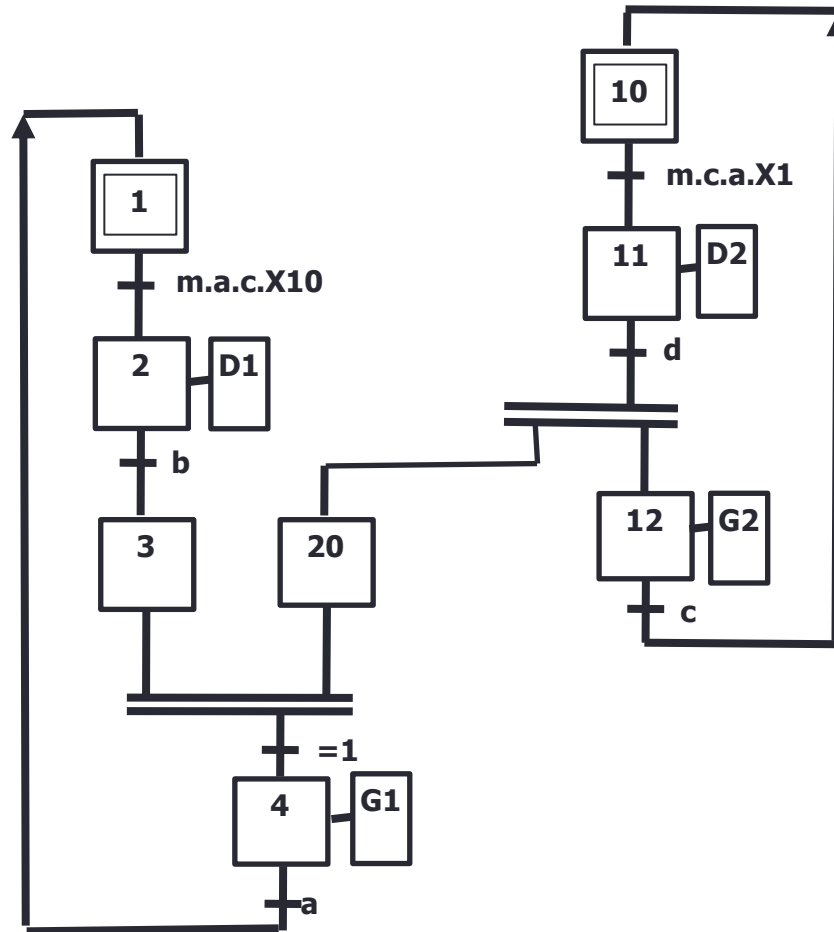
Mémorisation de passage



- **Entrées : m, a, b, c, d**
- **Sorties : G1, D1, G2, D2**

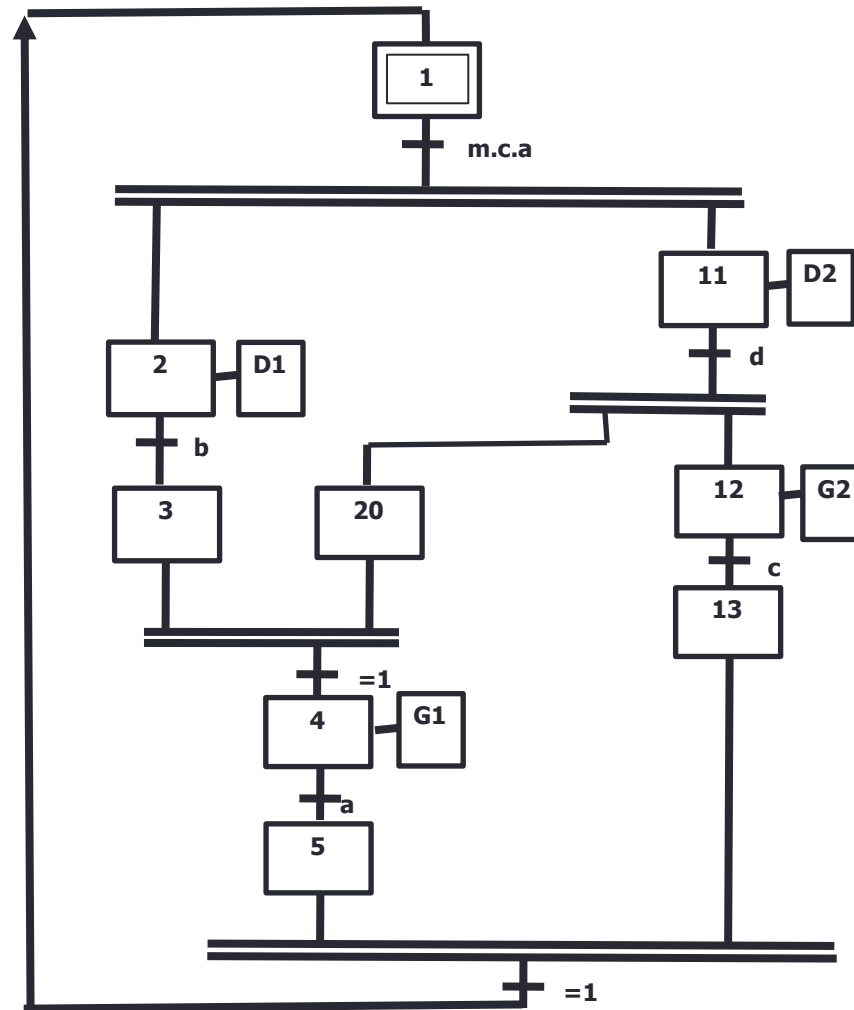
Mémorisation de passage

Solution 1



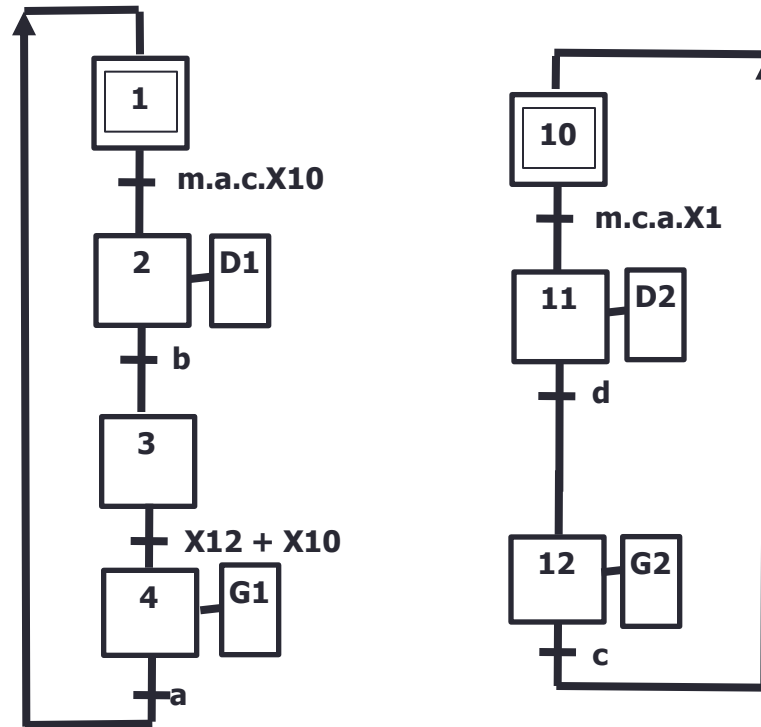
Mémorisation de passage

Solution 2



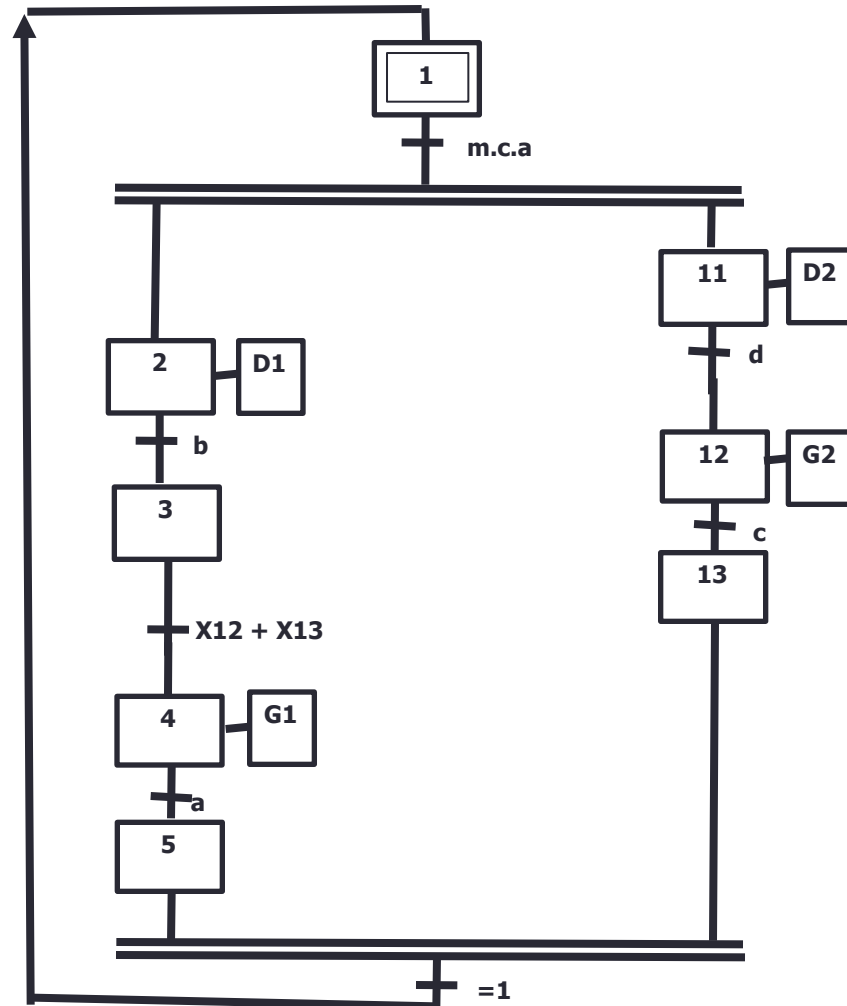
Mémorisation de passage

Solution 3



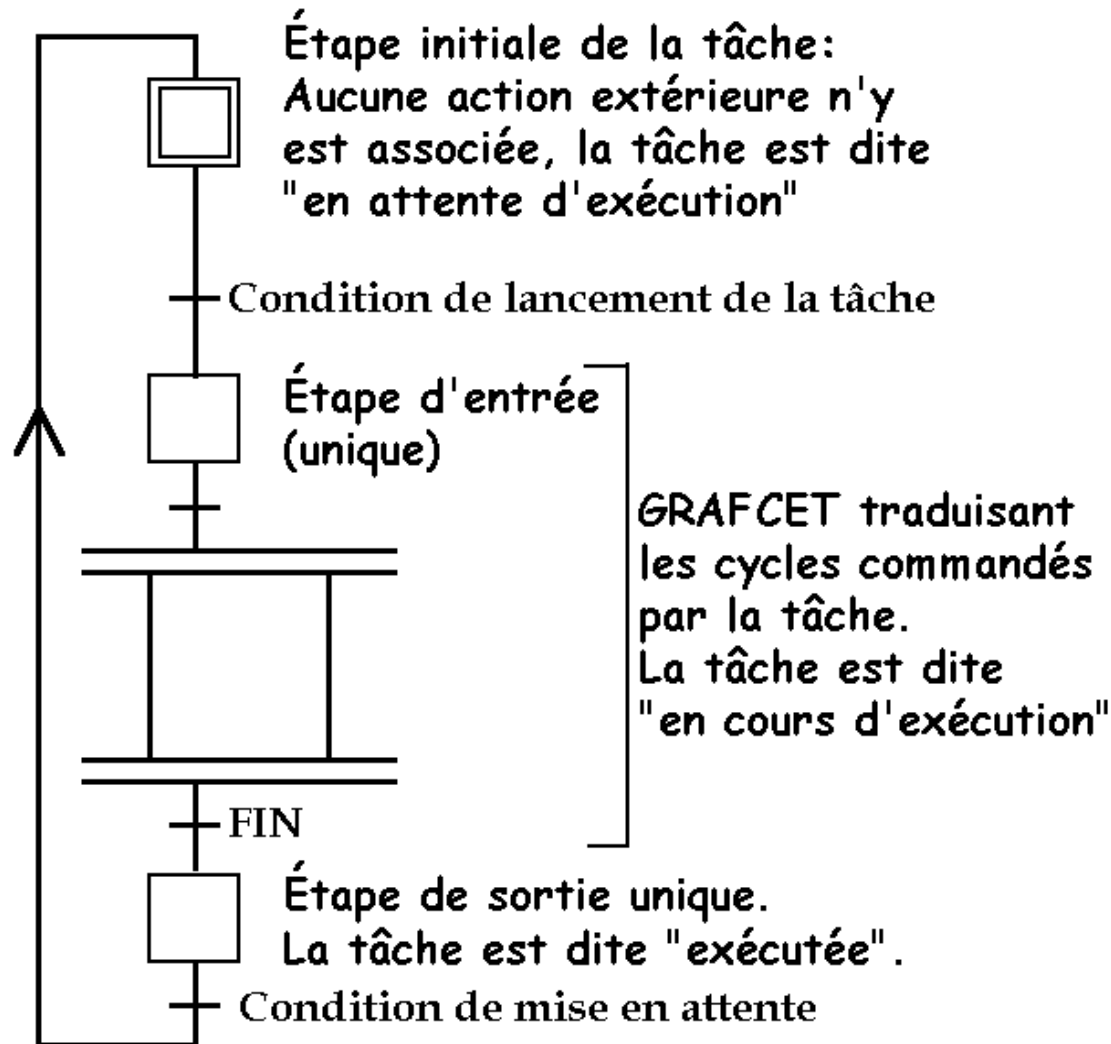
Mémorisation de passage

Solution 4



Grafcet de tâche

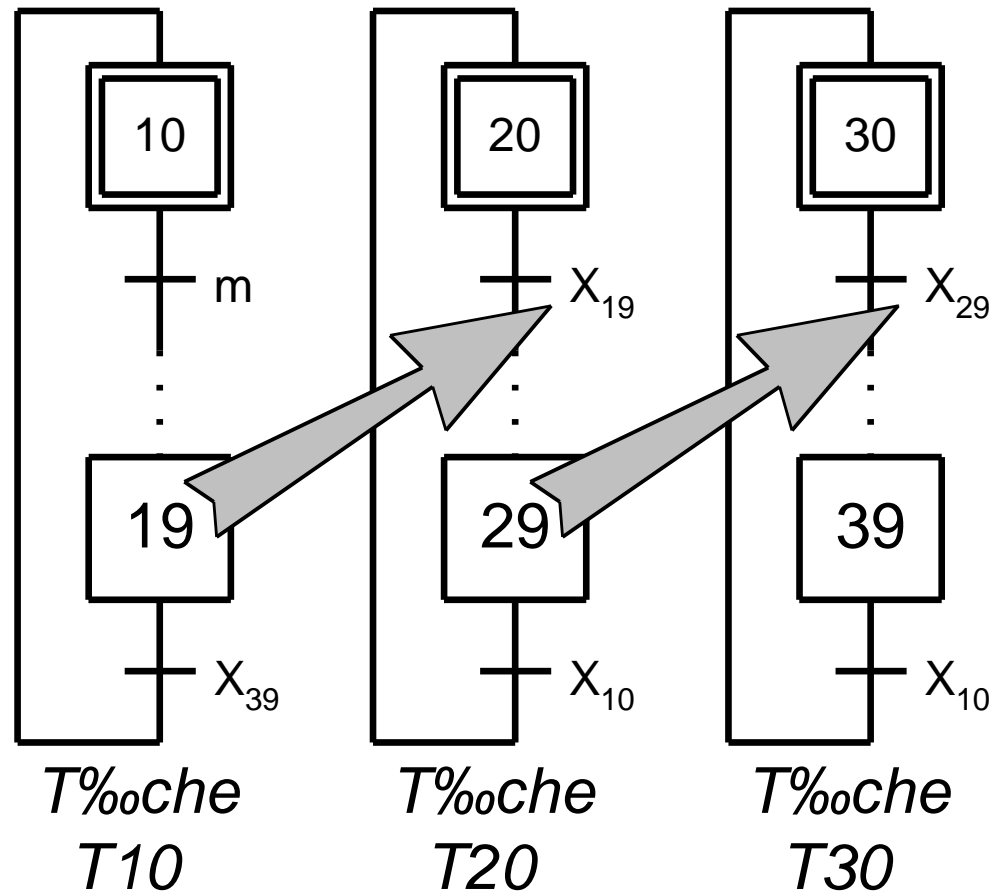
Notion de Grafcet de tâche



Grafctet de tâche

Coordination horizontale

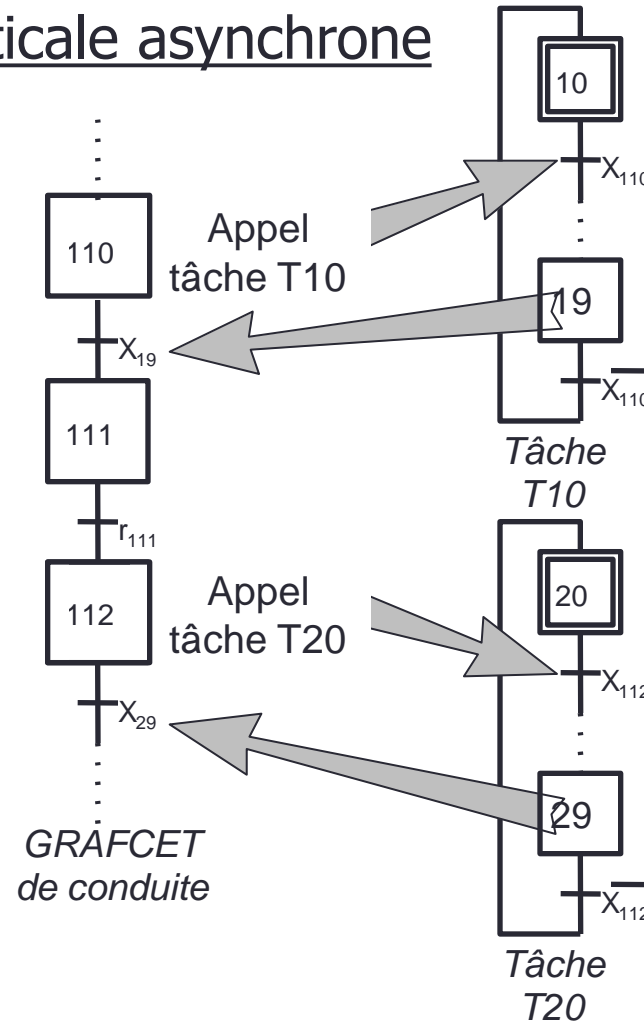
1 seule tâche à la fois



Synchronisation de Graficets

Coordination verticale asynchrone

GRAF CET de conduite
GRAF CET esclaves

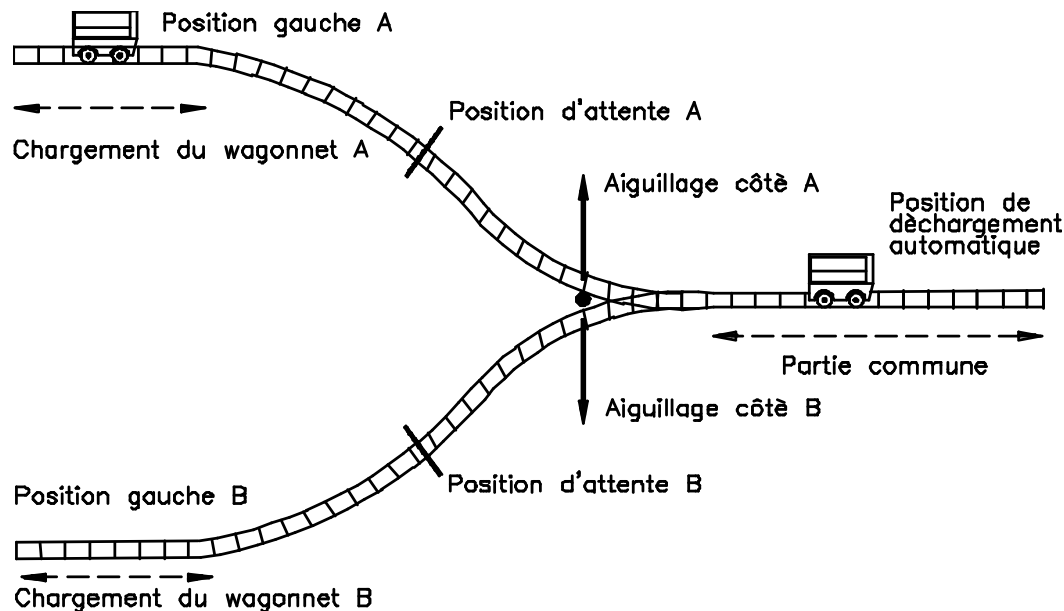


Séquences exclusives

Partage de ressource – gestion des problèmes d'arbitrage

- **Exemple : Déchargement de deux wagonnets**

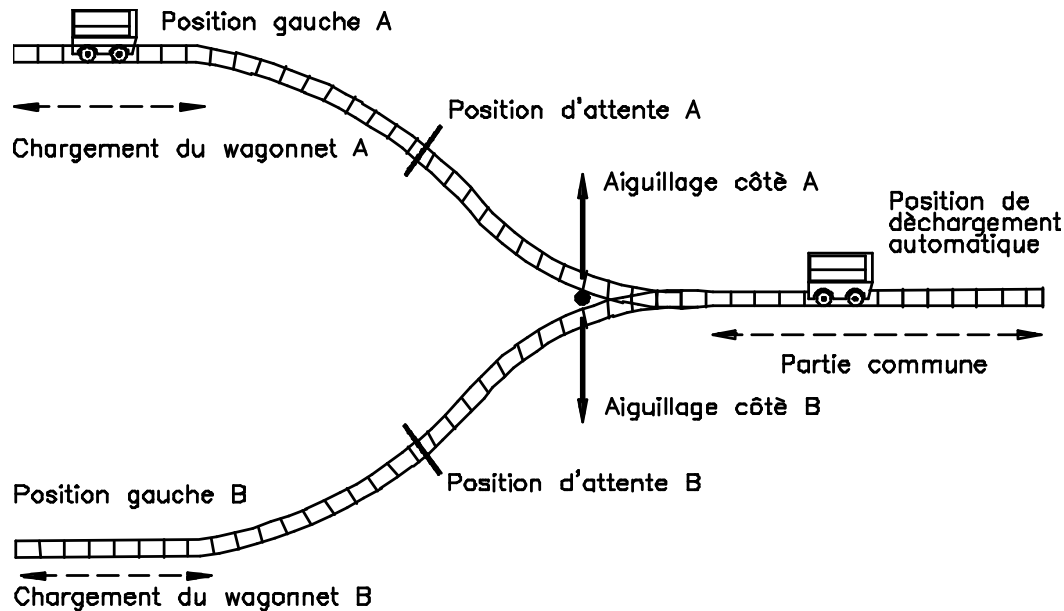
Deux wagonnets alimentent le bassin de chargement d'un haut fourneau en empruntant une voie commune.



Séquences exclusives

- Le cycle correspondant à un chariot est le suivant :
 1. Dès que l'opérateur donne l'ordre « départ cycle », le wagonnet considéré effectue automatiquement, dans la zone de chargement, les différents dosages choisis par l'opérateur.
- Le wagonnet se dirige ensuite vers la partie commune et il s'arrête à une position d'attente si celle-ci est occupée, sinon il continue directement en positionnant l'aiguillage sur la position correcte.
- Arrivé à la position de déchargement automatique, il attend 10 secondes avant de retourner à sa position initiale.
- Chaque déchargement d'un wagonnet est comptabilisé en vue d'une gestion journalière.

Séquences exclusives

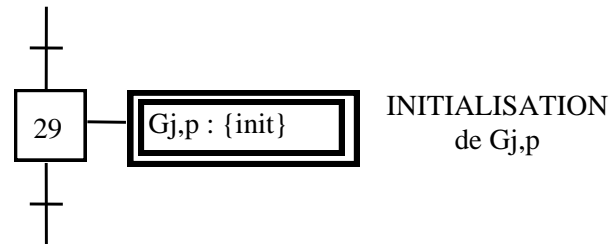


- **Entrées** : dcy A, position gauche A, dosage A terminé, position d'attente A, position de déchargement, aiguillage côté A, dcy B, position gauche B, dosage B terminé, position d'attente B, aiguillage côté B
- **Sorties** : Dosage A, Marche avant A, Aiguillage côté A, Marche arrière A, Dosage B, Marche avant B, Aiguillage côté B, Marche arrière B,

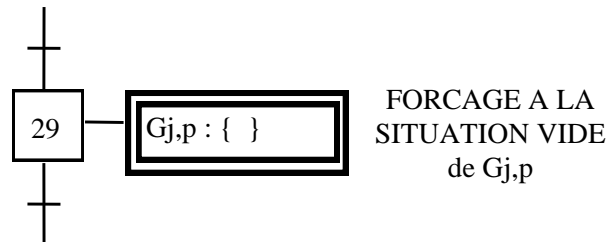
Forçages

L'ordre de forçage est représenté dans un double rectangle

1) Ordre d'initialisation : Les étapes initiales du grafcet partiel forcé sont activées, toutes les autres sont désactivées.

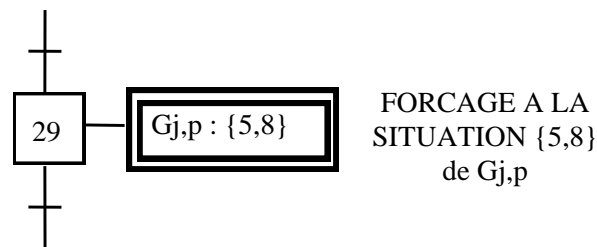


2) Forçage à la situation vide : Les étapes du grafcet partiel forcé sont toutes désactivées ; le redémarrage ne pourra être obtenu que par un autre ordre de forçage.



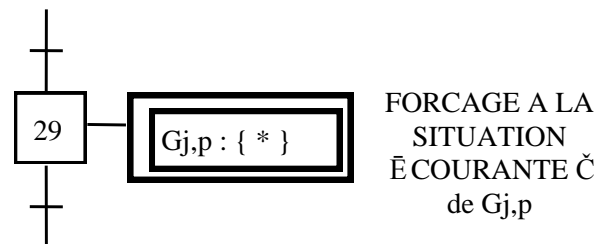
Forçages

3) Forçage à une situation donnée : Les étapes du grafcet partiel forcé dont les repères sont indiqués entre accolades sont activées, toutes les autres sont désactivées.



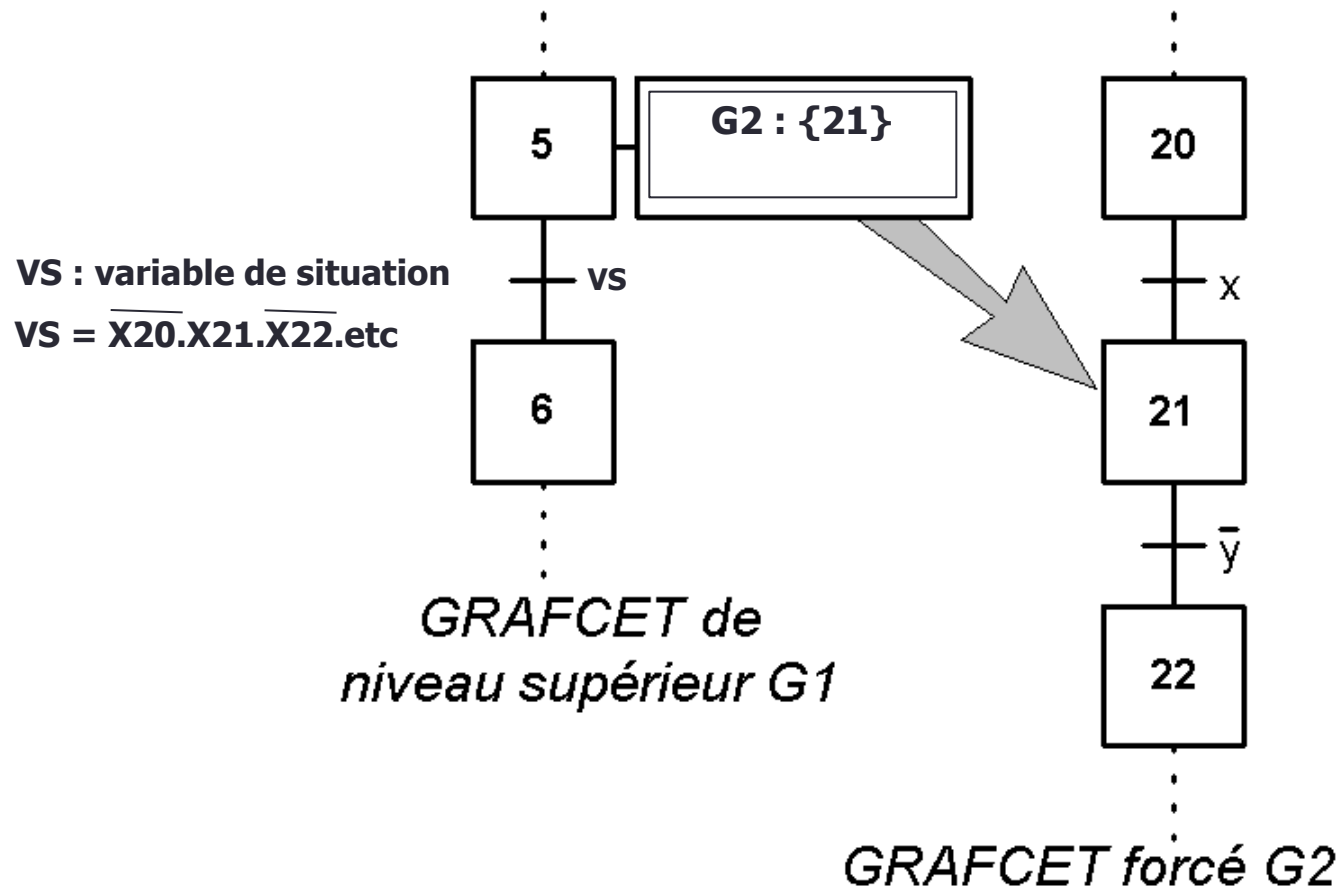
4) Forçage à la situation « courante » : Le grafcet partiel forcé garde la situation qu'il avait au moment où l'ordre de forçage est émis. Une étoile entre parenthèses symbolise la situation forcée.

figeage



Forçages

Exemple de forçage à une situation donnée

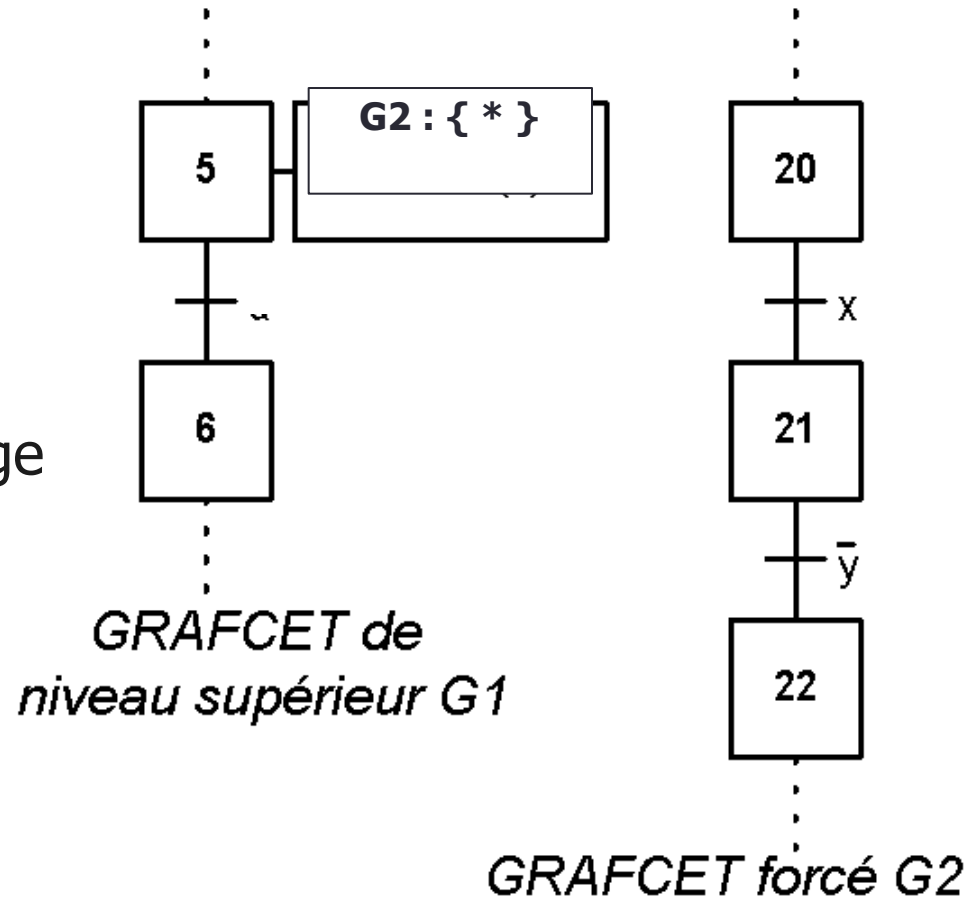


Forçages

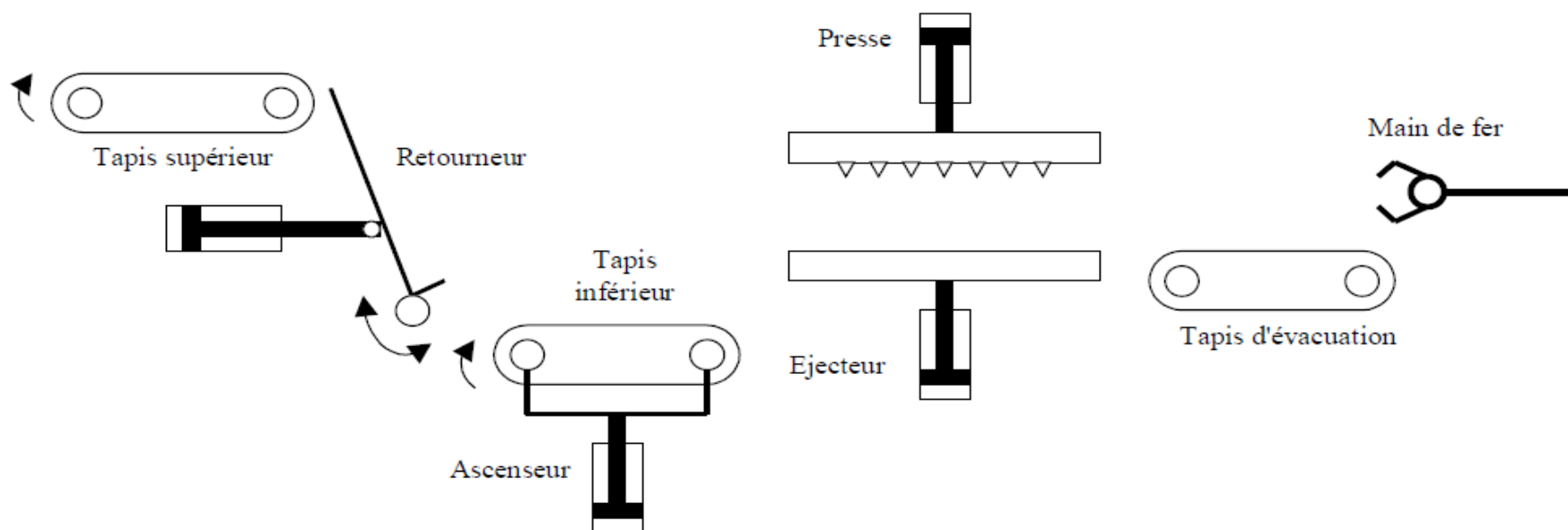
Exemple de forçage à la situation courante (figeage)

ATTENTION:

Les actions se poursuivent pendant le figeage



TD : Presse à emboutir avec alimentation et évacuation



Une presse à emboutir est alimentée par un dispositif formé par un tapis supérieur amenant les pièces dans un retourneur qui les retourne sur un tapis inférieur. Ce dernier élève les pièces jusqu'au niveau de la presse grâce à un ascenseur. Après emboutissage, la main de fer prend la pièce, qui a été préalablement surélevée par l'éjecteur, pour la déposer sur le tapis d'évacuation.

Capteurs

Dcy:	départ cycle,
pp :	présence pièce,
psp :	pièce sous presse,
pr :	pièce dans retourneur,
ab, ah :	ascenseur en bas, en haut,
rav, rar :	retourneur en AV /AR,
ph, pb :	presse en haut, en bas,
eb, eh :	éjecteur en bas, en haut,
mfar, mfav :	main de fer en AR / AV,
po, pf :	pince ouverte, pince fermée,

Actionneurs

RTS, RTI :	Rotation tapis supérieur / inférieur
MA, DA :	Montée / Descente ascenseur,
R, /R :	Retournement, inverse retournement,
MP, DP :	Montée / Descente presse,
DE, ME :	Descente / Montée éjecteur,
AMF, RMF :	Avance / Recul main de fer,

Choix de la technologie de mise en œuvre d'un Grafcet

Dans toute application, le Grafcet développé doit être matérialisé lors de la mise en œuvre. On peut matérialiser un Grafcet en utilisant

- un automate programmable industriel (API),
- l'aide de séquenceurs,
- de circuits logiques, etc.

Le choix d'une technologie dépend de plusieurs critères :

- Critères fonctionnels
- Critères technologiques
- Critères opérationnels
- Critères économiques

Les solutions envisagées sont :

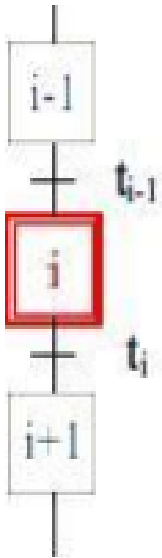
- La logique câblée :
 - La technologie électromagnétique (relais)
 - La technologie électronique (opérateurs logiques)
 - La technologie pneumatique
- La logique programmée

Gestion des modes Marche /Arrêt et des arrêts d'urgences

- A l'initialisation du GRAFCET, toutes les étapes autres que les étapes initiales sont désactivées. Seules sont activées les étapes initiales.
- Soit la variable Init telle que :
 - Init = 1 : initialisation du GRAFCET : Mode ARRÊT
 - Init = 0 : déroulement du cycle: Mode MARCHÉ
- Soient les variables Arrêt d'Urgence (AUdur et AUdoux) telles que :
 - AUDur = 1 :Désactivation de toutes les étapes,
 - AUDoux = 1 : Désactivation des actions, les étapes restant actives .

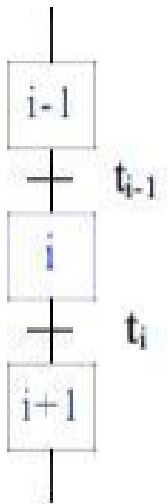
Généralisation :

Équation d'une étape i initiale :



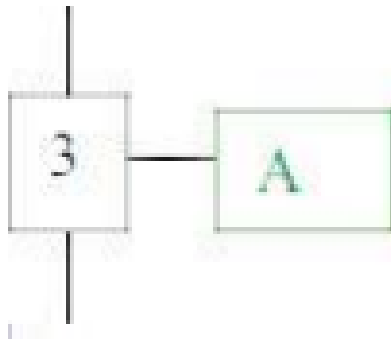
CAXi	CDXi	Equation de Xi
$X_{i-1}t_{i-1} + Init$	$X_{i+1} \overline{Init}$	$X_i = \left(CAX_i + \overline{CDX}_i \cdot X_i + Init \right) \overline{AUdur}$

Équation d'une étape i non initiale :



CAX_i	CDX_i	Equation de X_i
$X_{i-1} t_{i-1} \overline{Init}$	$X_{i+1} \cdot + Init$	$X_i = \left(CAX_i + \overline{CDX}_i \cdot X_i \right) \cdot \overline{Init} \cdot \overline{AUdur}$

Équation des actions



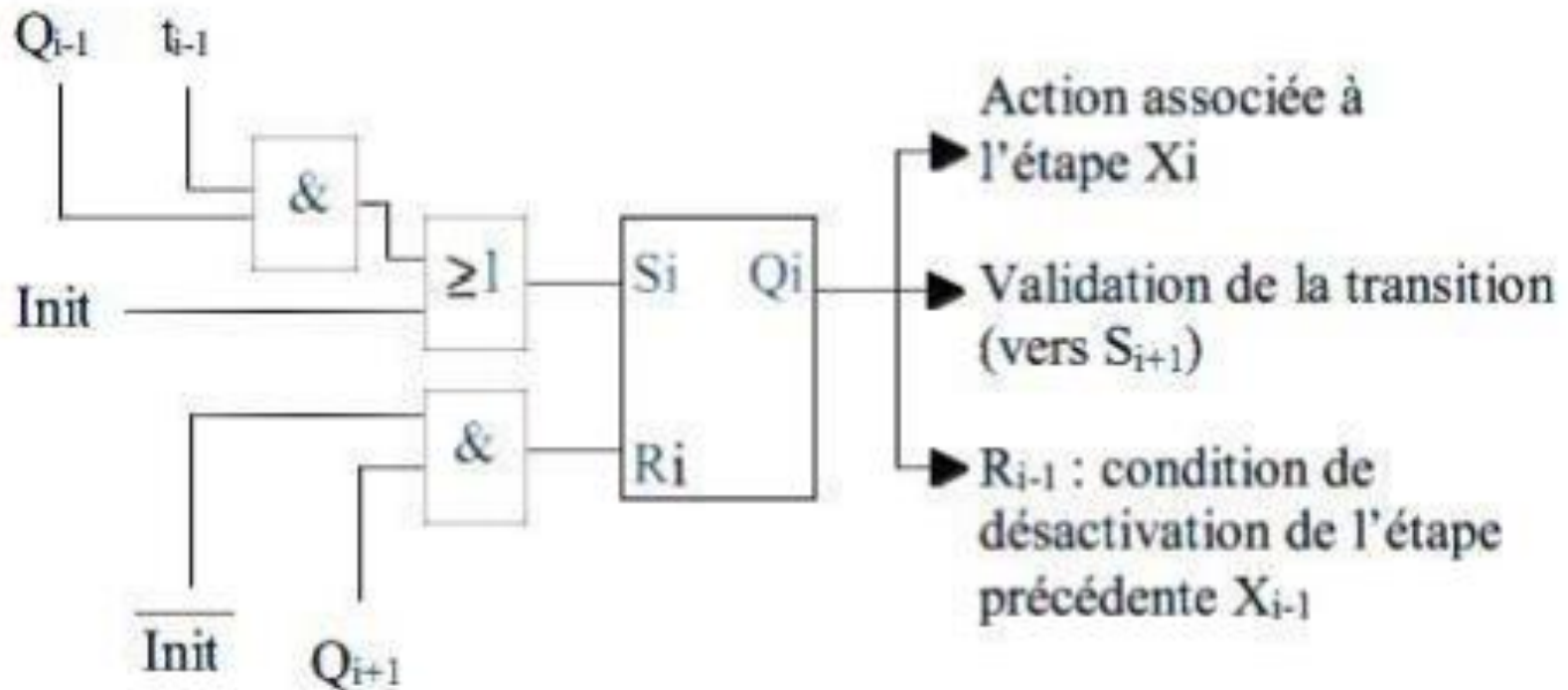
$$A = \overline{X3.A} U \text{doux}$$

Matérialisation par câblage

- Câblage d'une étape initiale :

$$CAX_i = X_{i-1} \cdot t_{i-1} + Init$$

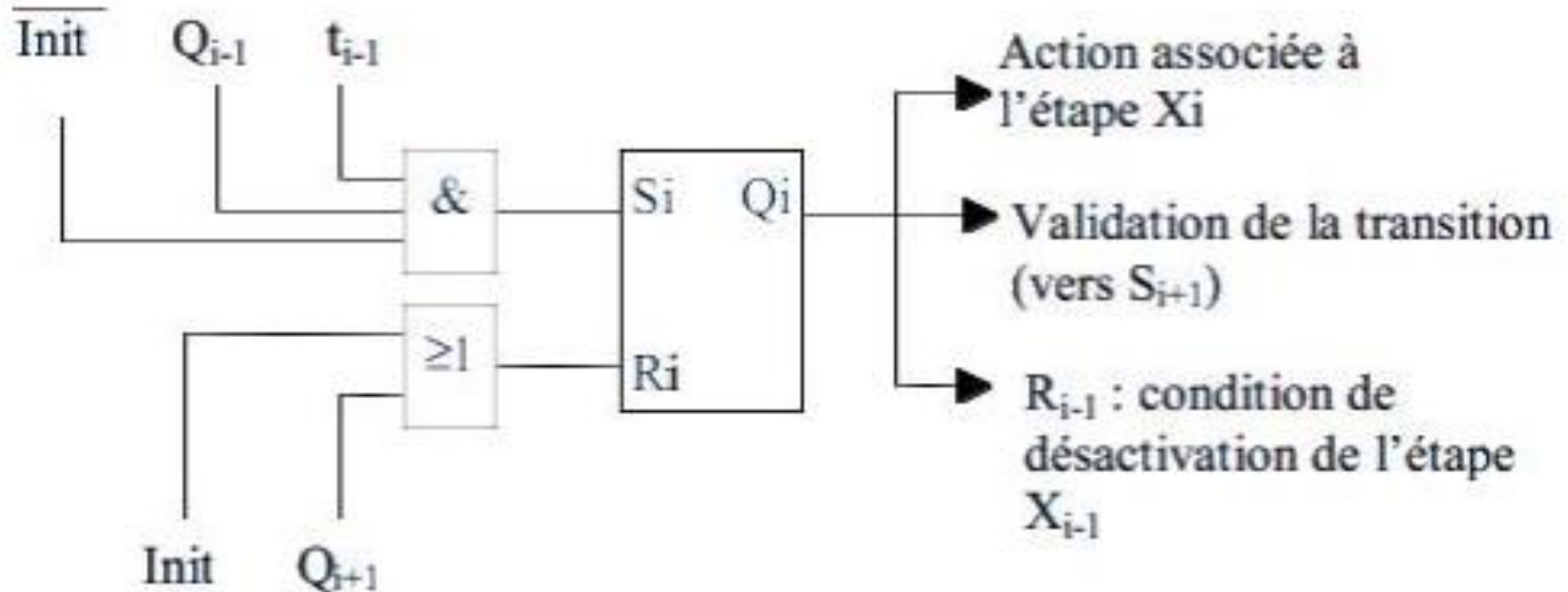
$$CDX_i = X_{i+1} \cdot \overline{Init}$$



- Câblage d'une étape non initiale :

$$CAX_i = X_{i-1} \cdot t_{i-1} \cdot \overline{Init}$$

$$CDX_i = X_{i+1} + Init$$



Exemple : GRAFCET à séquence unique:

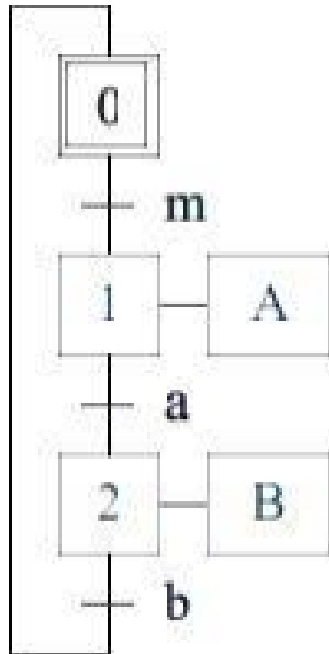
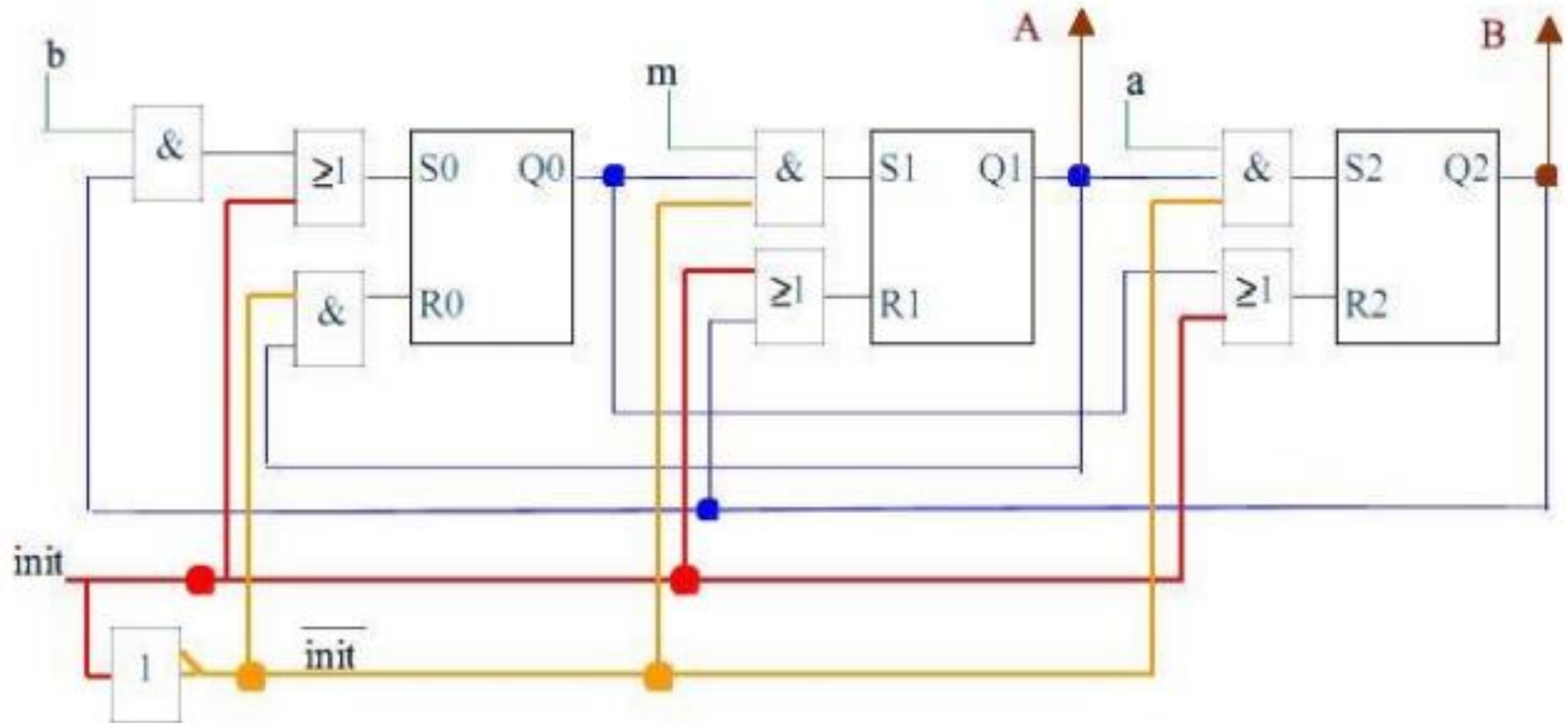


Table des conditions d'activation et de désactivation des étapes :

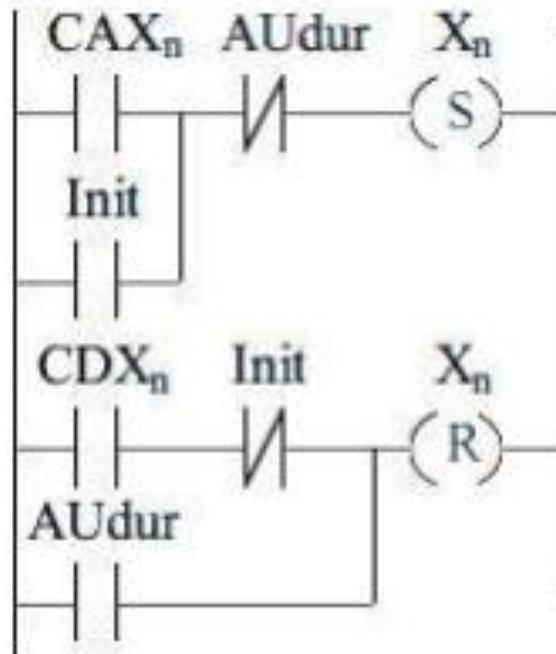
X_n	CAX_n	CDX_n
0	$X_2.b + Init$	$X_1.\overline{Init}$
1	$X_0.m.\overline{Init}$	$X_2 + Init$
2	$X_1.a.\overline{Init}$	$X_0 + Init$

Câblage du Grafcet

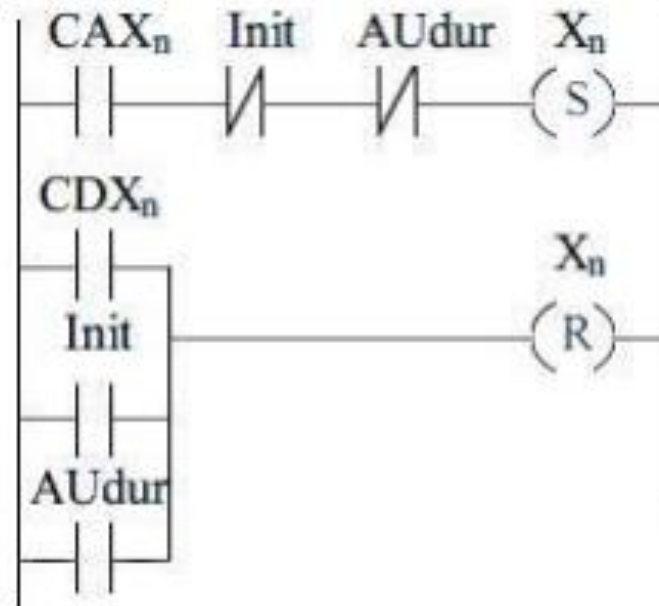


Programmation du Grafcet dans un langage à contacts (Ladder)

Étape initiale :



Étape NON initiale :



Action associée à l'étape :

