
4.

GRAFCET OUTIL D'ANALYSE

Pierre Duysinx

Université de Liège

Année académique 2021-2022

GRAF CET : UN OUTIL D'ANALYSE

Approche progressive

- **Rédaction du cahier des charges:** description claire, précise, sans ambiguïté ni omission du rôle et des performances de l'équipement

- Diviser la description en plusieurs niveaux ou points de vue:
 - 1/ Point de vue système:
 - Description sous forme littérale du procédé, de la coordination et de l'évolution des différentes séquences.
 - Description abstraite **sans notions techniques.**
 - 2/ Point de vue partie opérative
 - Description sous forme d'actions fonctionnelles du comportement de la partie opérative.
 - Effectuer le **choix technologique.**
 - 3/ Point de vue partie commande
 - Description des échanges d'informations et d'ordres
 - **Langage technique**

Approche progressive

- Niveau 1: Point de vue système / Spécifications fonctionnelles
 - Description du comportement de la partie commande vis-à-vis de la partie opérative PO
 - Comprendre ce que l'automatisme doit faire
 - Caractériser les réactions de l'automatisme face aux infos de la PO
 - Définir les différentes fonctions, informations et commandes impliquées dans l'automatisation de la PO sans préjuger des technologies
 - Pas d'influence de la nature ni des caractéristiques des capteurs et actionneurs
 - La sécurité de fonctionnement est incluse dans les spécifications fonctionnelles

Approche progressive

- Niveau 2: Point de vue opérative / Spécifications technologiques
 - Préciser les spécifications technologiques des actionneurs, capteurs...
 - Spécification d'interface: insertion de l'automatisme dans le processus et son environnement
 - Préciser
 - La nature et les informations sur les capteurs et actionneurs
 - Les contraintes sur les capteurs (température, humidité...)

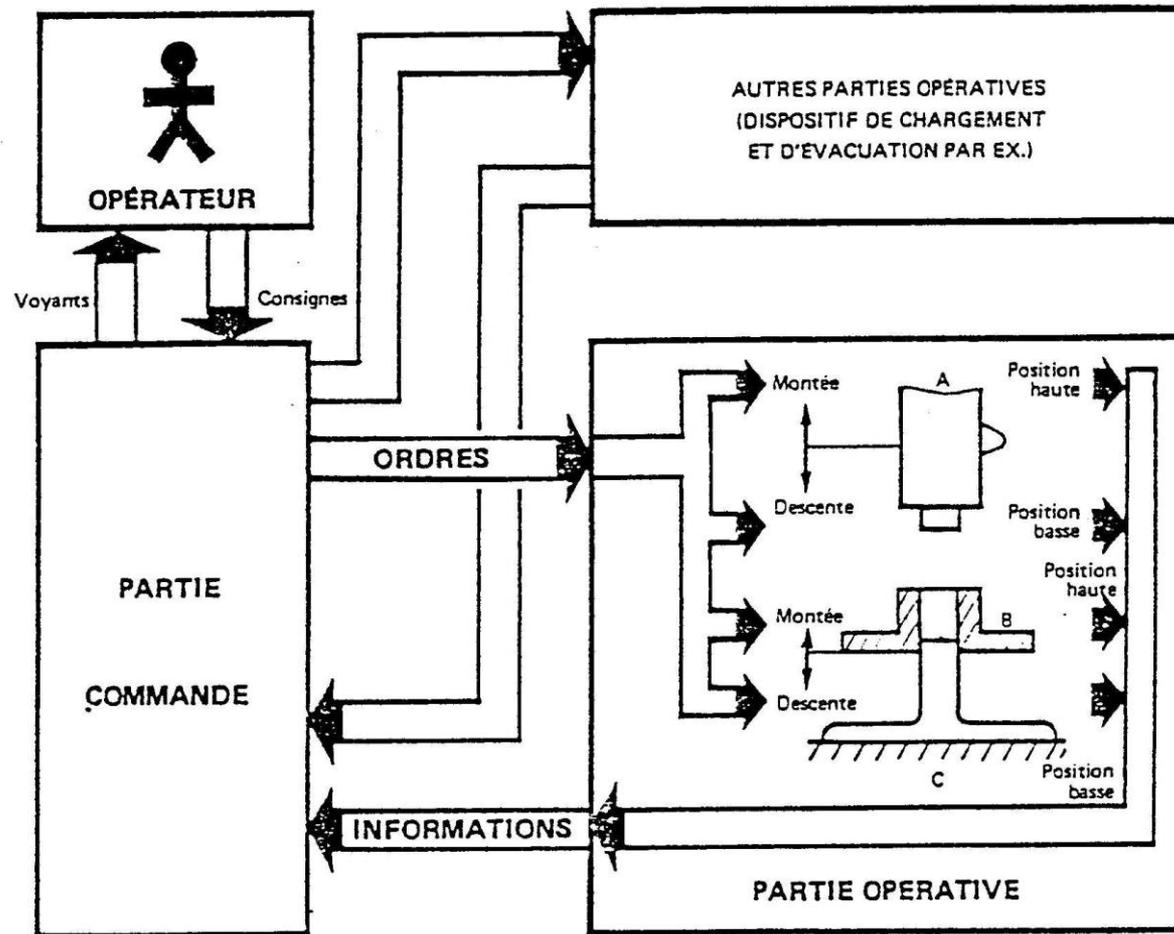
Approche progressive

- Niveau 3: Partie commande / Spécifications opérationnelles
 - Servir au suivi de fonctionnement de l'automatisme au cours de son existence
 - Définir les spécifications sur l'équipement une fois réalisé et en exploitation: fiabilité, pannes, disponibilité, modifications, maintenance, HMI...

GRAFCET: introduction

- Le GRAFCET est utilisé pour les différents niveaux de description
- Description de l'automatisme par un ensemble de:
 - ETAPES auxquelles on associe des ACTIONS
 - TRANSITIONS auxquelles on associe des RÉCEPTIVITÉS
 - LIAISONS ORIENTÉES reliant les étapes aux transitions et les transitions aux étapes
- Principes
 - La partie commande reste dans une étape tant que son comportement est constant.
 - On change d'étape lorsqu'elle est **réceptive** ce qui provoque le franchissement de la transition
 - Fonctionnement de la commande = **une succession alternée d'étapes et de transitions**

Exemple: presse



- Presse destinées à la fabrication de pièces à partir de poudre comprimée

Exemple: presse

- Partie opérative:
 - Poinçon inférieur fixe C
 - Poinçon supérieur A et une partie mobile B
 - Sous ensemble de mise en place de la matière
 - Sous ensemble d'évacuation de la pièce

Exemple: presse

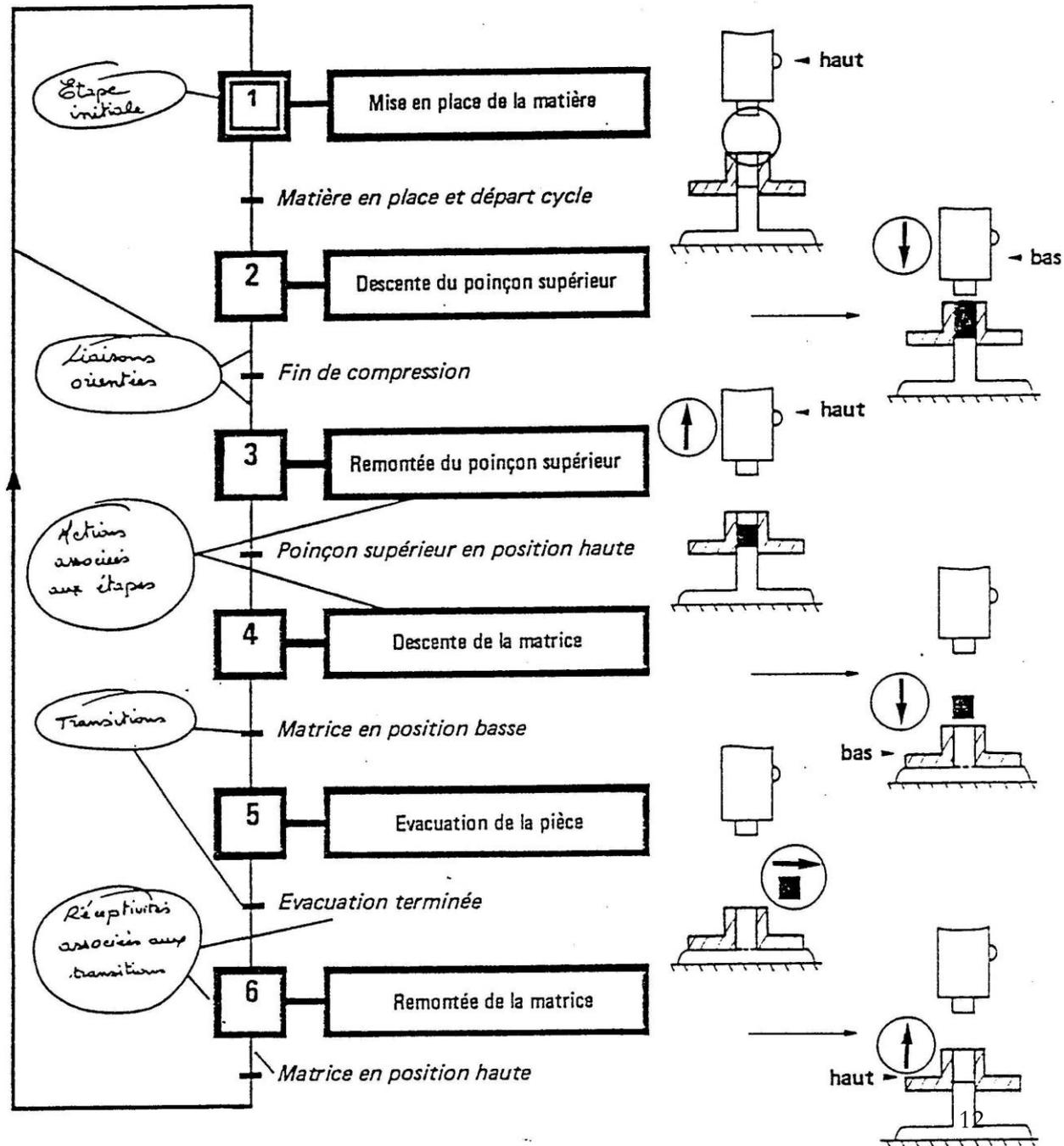
- Fonctionnement du cycle
 - Matrice en haut de sa course. Poinçon inférieur engagé délimite un espace suffisant pour recevoir de la matière. Le poinçon supérieur est alors dans sa position haute pour permettre l'introduction de la poudre
 - Lors la matière pulvérulente est en place, le poinçon supérieur descend, comprime la matière puis remonte en position haute
 - La matrice descend jusqu'à ce que le poinçon inférieur affleure et libère la pièce. La pièce est ensuite évacuée
 - La matrice reprend sa place et le cycle peut recommencer.

Exemple: presse

- Etude de la partie commande – GRAFCET de niveau 1

Etape 1	;	action	:	mise en place de la matière
Transition 1-2	;	réceptivité	:	matière en place et départ cycle
Etape 2	;	action	:	descente du poinçon
Transition 2-3	;	réceptivité	:	fin de compression
Etape 3	;	action	:	remontée du poinçon
Transition 3-4	;	réceptivité	:	poinçon en haut
Etape 4	;	action	:	descente matrice
Transition 4-5	;	réceptivité	:	matrice en bas
Etape 5	;	action	:	évacuation de la pièce comprimée
Transition 5-6	;	réceptivité	:	pièce évacuée
Etape 6	;	action	:	remontée matrice
Transition 6-1	;	réceptivité	:	matrice en haut

Exemple



GRAFCET de niveau 1

Exemple: presse

- Etude de la partie commande – GRAFCET de niveau 2
 - Il convient de préciser les choix technologiques des actionneurs et des capteurs

- Dans cet exemple
 - Pas de procédure d'arrêt d'urgence
 - Mise en place manuelle de la matière par un opérateur. Un voyant vert est allumé pendant la durée de mise en place. A la fin, l'opérateur autorise la suite des opérations en poussant sur un bouton poussoir d
 - Les mouvements du poinçon supérieur et de la matrice sont réalisés par des vérins hydrauliques double effet. Les positions hautes et basses sont contrôlées par des capteurs de fin de course a0, a1, b1, b0
 - L'évacuation de la pièce est commandée par un jet d'air actionné par une électrovanne E

Exemple: presse

■ Etude de la partie commande – GRAFCET de niveau 2

ORDRES

vers le milieu extérieur et l'opérateur

V : voyant "Prêt"

commande des actionneurs

a + : descente poinçon

a - : remontée poinçon

b - : descente matrice

b + : remontée matrice

E : évacuation

lancement de temporisations

LT1 : lancement temporisation d'évacuation

INFORMATIONS

déroulement du cycle

d : autorisation de départ cycle

fin de course des actionneurs

a1 : position basse du poinçon

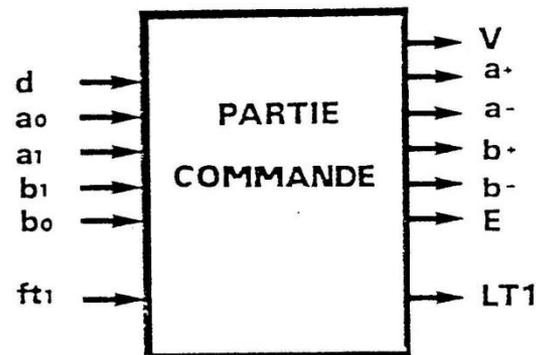
a0 : position haute du poinçon

b0 : position basse de la matrice

b1 : position haute de la matrice

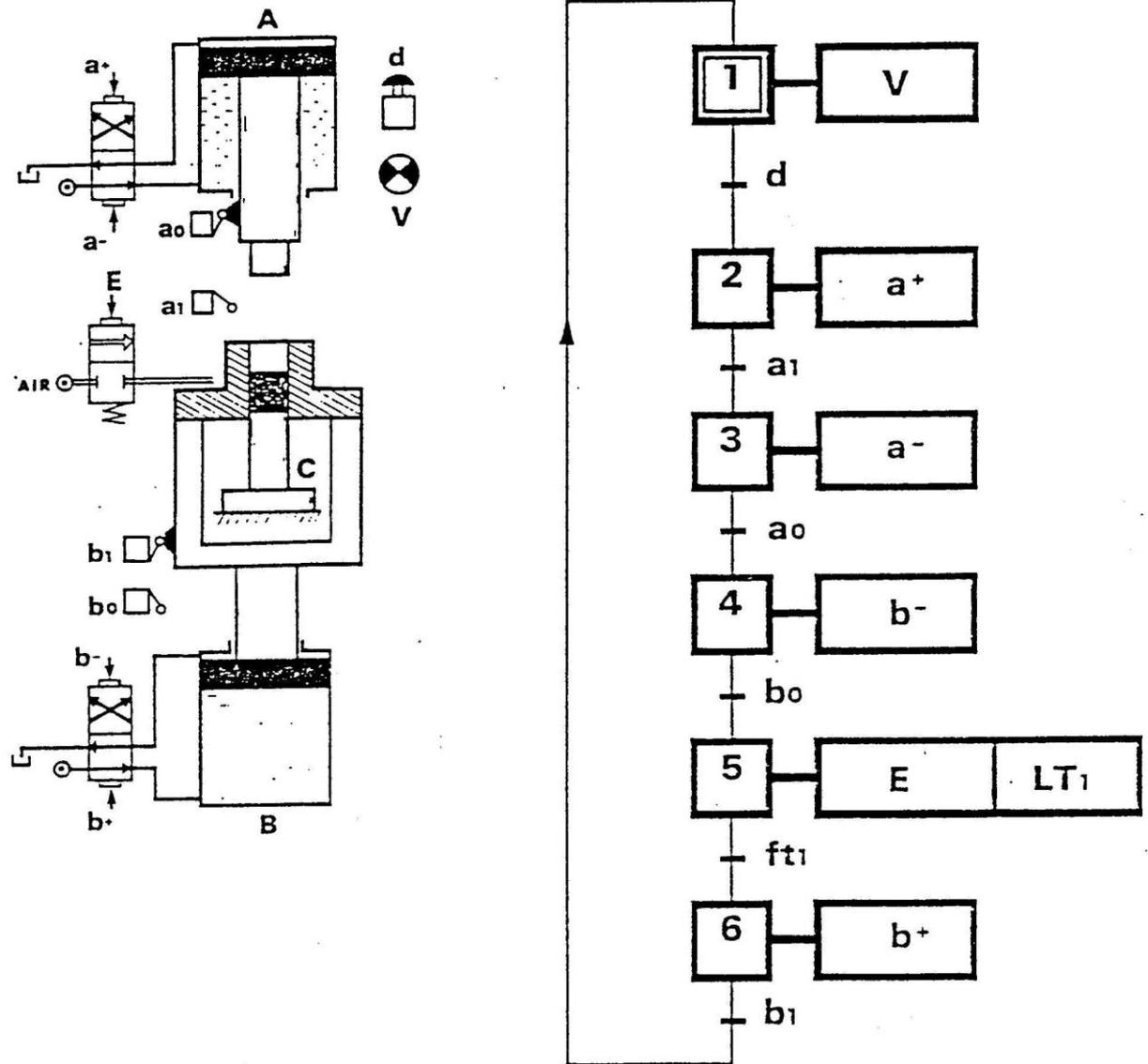
fin de temporisation

ft1 : fin de temporisation d'évacuation



Exemple

GRAF CET de
niveau 2



GRAFCET: METHODOLOGIE DE GESTION D'UN PROJET

GRAFCET

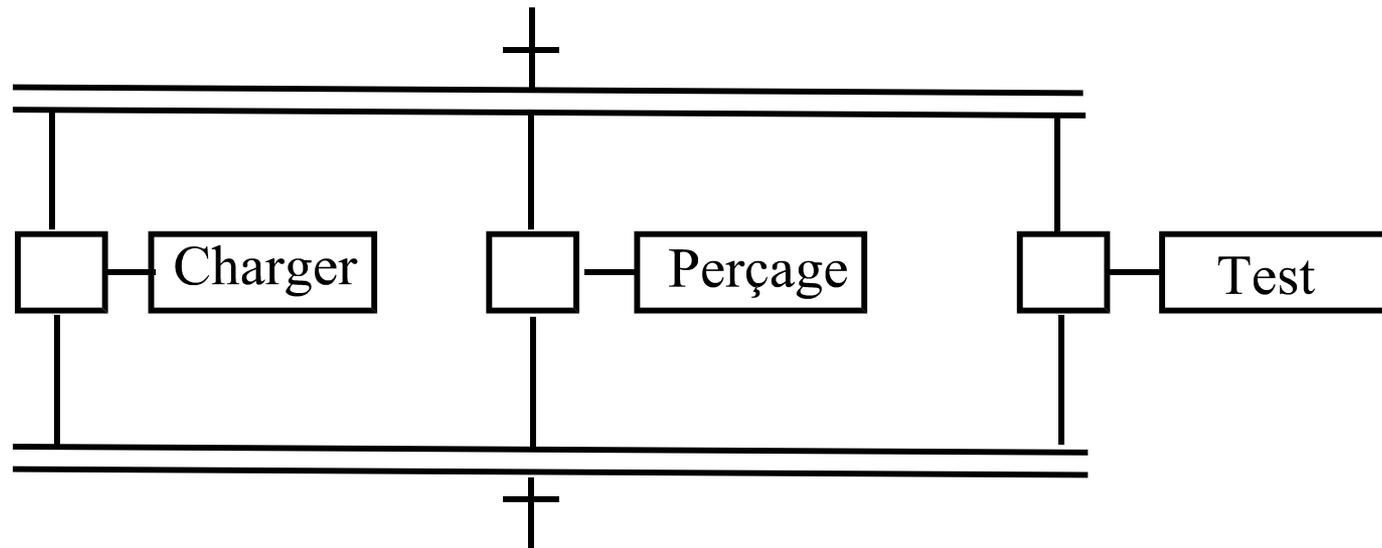
- METHODOLOGIE DE GESTION D'UN PROJET
- Structuration
 - Approche top down
 - Utilisation de sous grafkets
- Synchronisation : gestion du multitâche
 - Le mode de représentation GRAFCET autorise le lancement d'activités en parallèle et se déroulant de manière indépendante
 - Nécessité de resynchroniser ces activités
- Protection des ressources communes
 - Eviter les accès simultanés
- Transposition du GRAFCET en langage automate

GRAFCET

- METHODOLOGIE DE GESTION D'UN PROJET
- Structuration
 - Approche top down
 - Utilisation de sous grafcets
- Synchronisation : gestion du multitâche
 - Le mode de représentation GRAFCET autorise le lancement d'activités en parallèle et se déroulant de manière indépendante
 - Nécessité de resynchroniser ces activités
- Protection des ressources communes
 - Eviter les accès simultanés
- Transposition du GRAFCET en langage automate

GRAFCET: Structuration

- Pour garder au GRAFCET son potentiel de clarté: il faut le structurer
- Aborder la conception d'un automatisme dans une approche top-down: du général vers le particulier
- Utiliser des étapes pouvant être décomposées en des GRAFCETS eux-mêmes

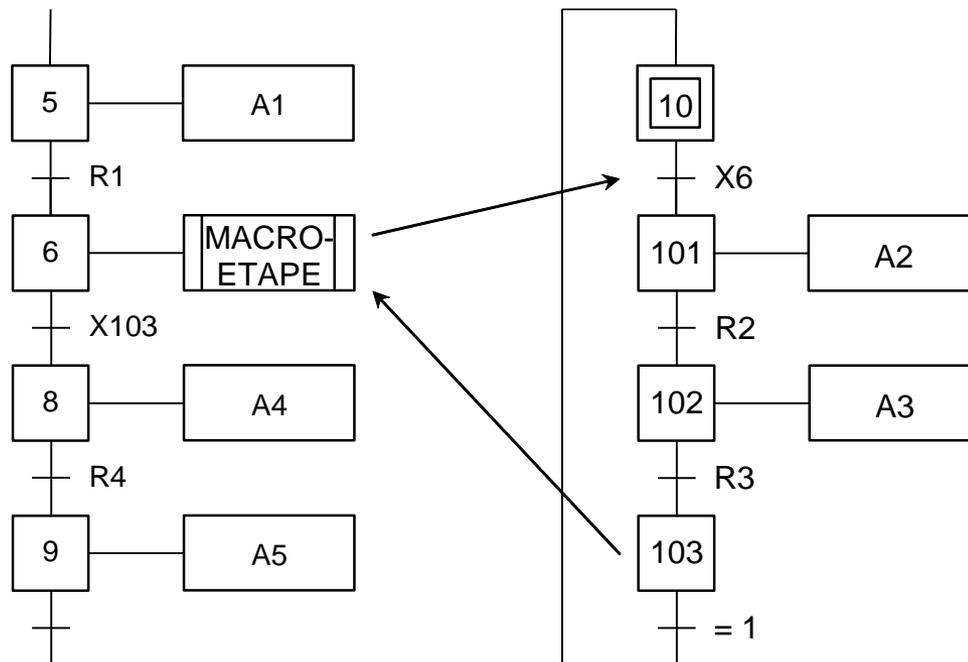


GRAFCET: Structuration

- Intérêt de l'approche TOP-DOWN
 - Plusieurs niveaux de description ⇒ facilite la description
 - Niveaux supérieurs compréhensibles par des non spécialistes
 - Outil de dialogue entre différents métiers
 - Répartition du travail de développement
 - Entre plusieurs équipes
 - Test et maintenance :
 - Identifier les blocs origines du problème
 - Objectif: à chaque niveau, on peut avoir un GRAFCET sur une page ou à l'écran

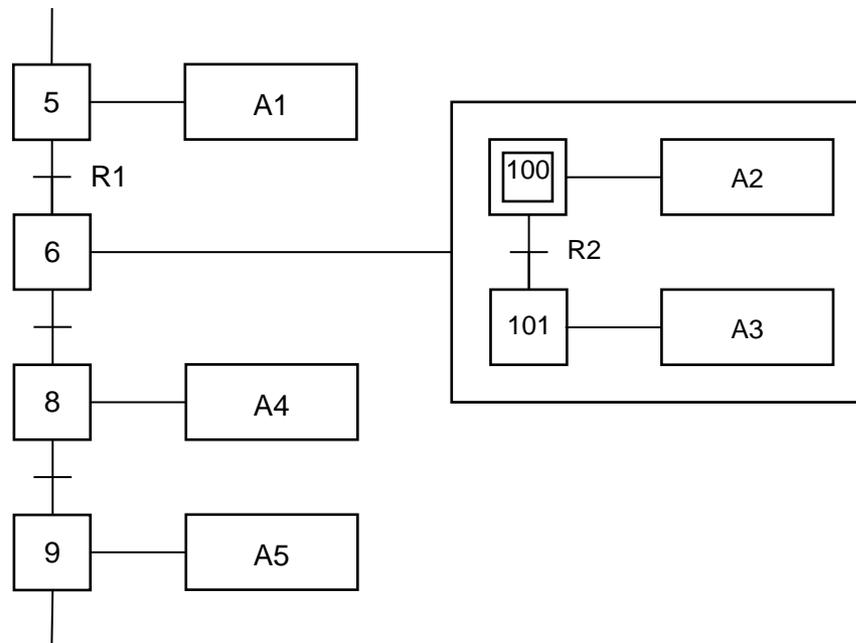
GRAFCET: Structuration

- Utilisation des macro étapes (rappel)



GRAF CET: Structuration

- Utilisation des actions Grafcet (rappel)

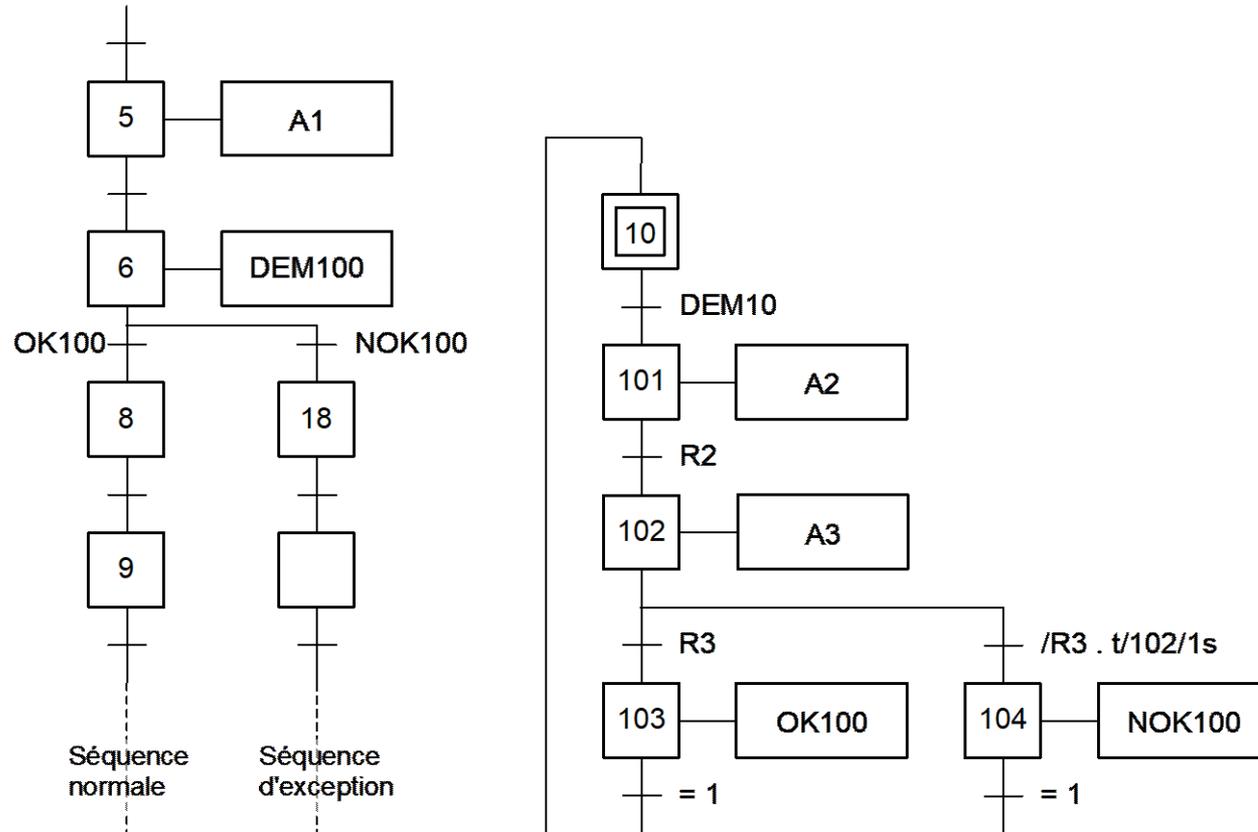


GRAFCET: Structuration

- Utilisation du **principe du client serveur**
 - Consiste à utiliser une action GRAFCET principal pour lancer le sous GRAFCET
 - Le GRAFCET principal fait une demande de service au sous GRAFCET
 - Une fois sa tâche terminée le sous GRAFCET émet un compte-rendu d'activité (bit de bonne fin) qui permet de poursuivre la séquence

- Avantages de la procédure client serveur
 - Indépendance vis-à-vis de la numérotation des étapes
 - Facilité de tester séparément les SOUS-GRAFCETS
 - Facilité d'introduire des fonctions de diagnostic (variable de bonne fin ou non)

GRAF CET: Structuration

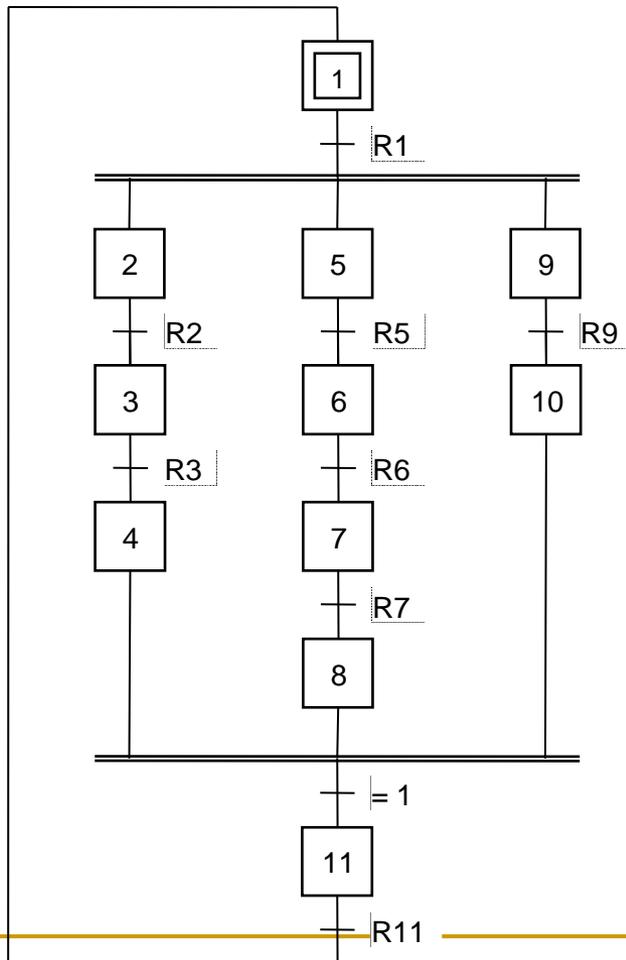


Principe du client serveur pour la structuration des GRACETs

GRAFCET

- METHODOLOGIE DE GESTION D'UN PROJET
- Structuration
 - Approche top down
 - Utilisation de sous grafkets
- Synchronisation : gestion du multitâche
 - Le mode de représentation GRAFCET autorise le lancement d'activités en parallèle et se déroulant de manière indépendante
 - Nécessité de resynchroniser ces activités
- Protection des ressources communes
 - Eviter les accès simultanés
- Transposition du GRAFCET en langage automate

GRAFCET: Synchronisation



- **Synchronisation explicite**
 - Regroupement de branches parallèles sur une double barre horizontale avec une réceptivité unique.
 - S'applique **lorsque les activités parallèles sont si étroitement liées** qu'une représentation graphique explicite de leurs interactions est indispensable.
 - Lorsque les activités sont faiblement couplées, on lui préfère des GRAFCETs indépendants pour des raisons de clarté. On les synchronise implicitement par variables d'étape.

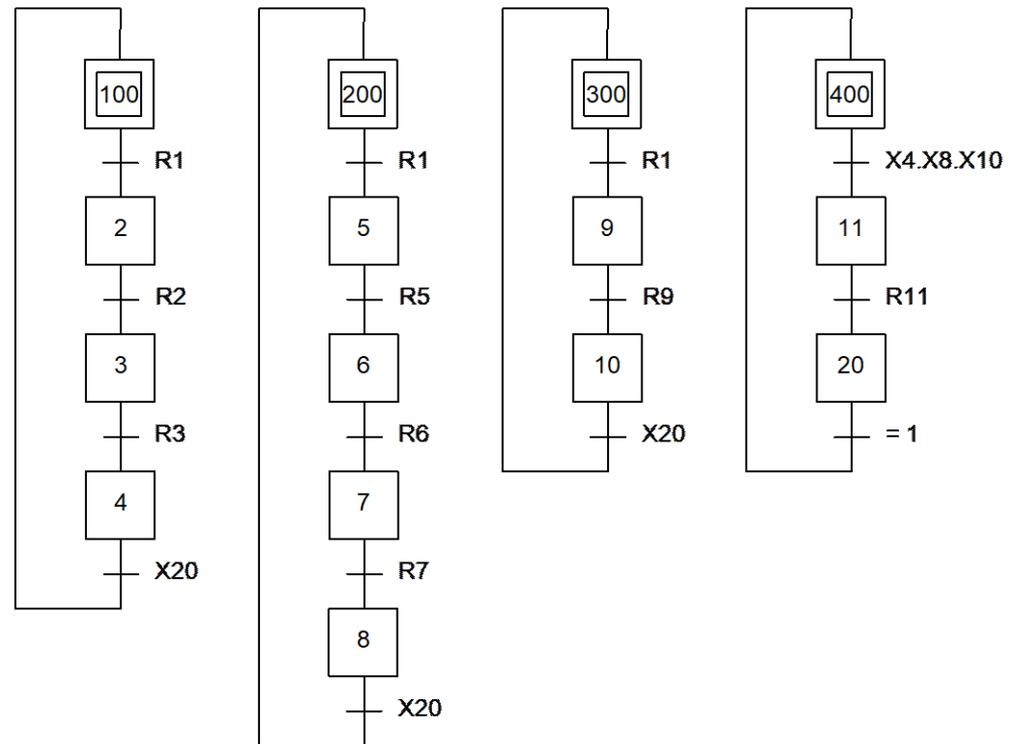
GRAFCET: Synchronisation

■ Synchronisation implicite horizontale:

- Echange de variables entre GRAFCETs de même niveau
- Les activités sont décrites par des GRAFCETs graphiquement indépendants
- Les activités sont synchronisées par des bits d'étapes croisés qui servent de réceptivités dans les autres GRAFCETs

Synchronisation d'activités parallèles : méthode implicite

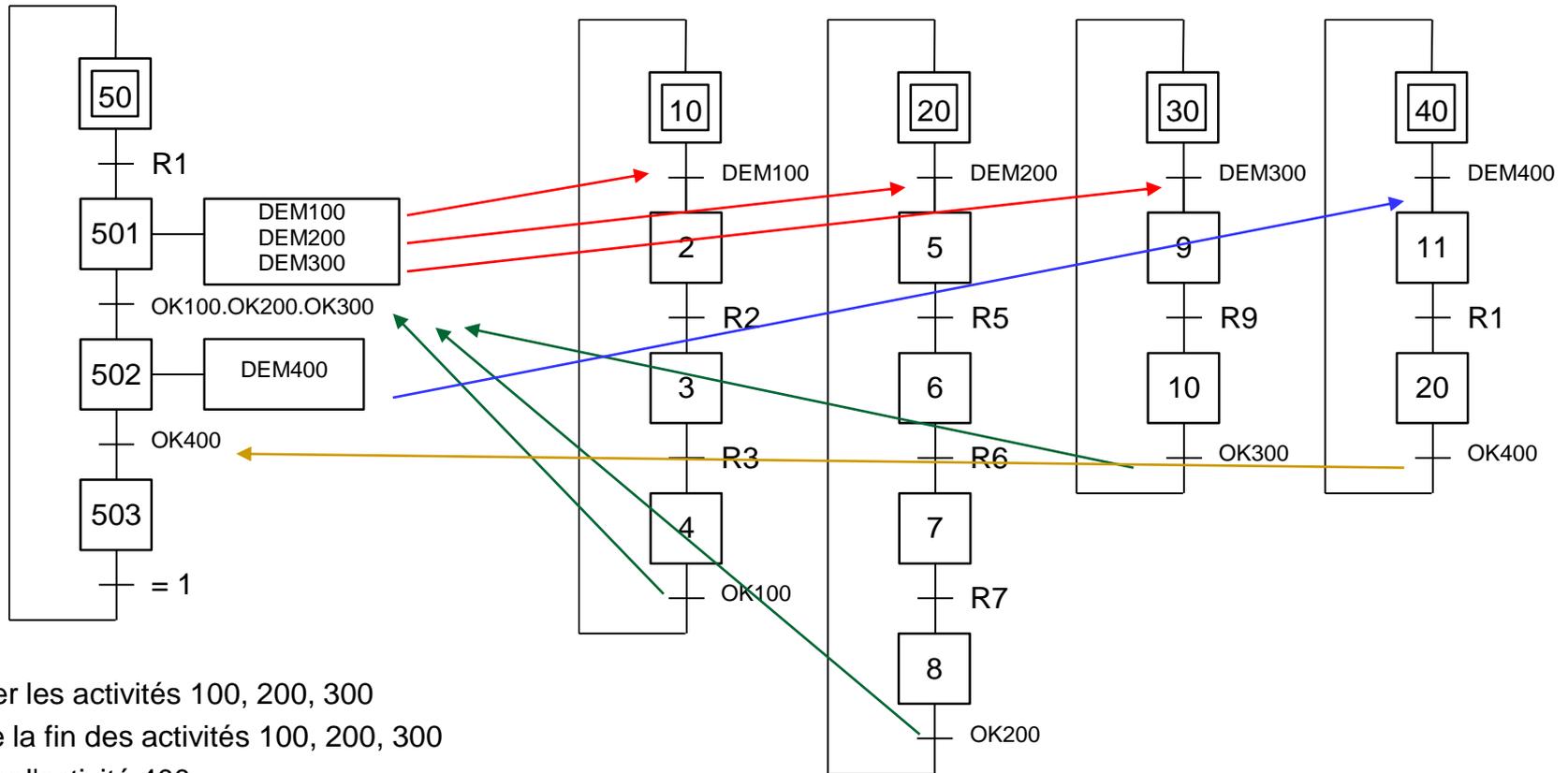
SYNCHRONISATION HORIZONTALE



GRAFCET: Synchronisation

- **Synchronisation implicite verticale**
 - La synchronisation horizontale peut devenir difficile à suivre à cause des croisements de variables d'étape
 - Synchronisation verticale utilise un **GRAFCET supplémentaire de synchronisation** et utilise le principe du client serveur
 - Le **GRAFCET supérieur** contrôle l'exécution des activités
 - Les interactions ne se font qu'avec le GRAFCET de synchronisation

GRAFCET: Synchronisation



- Démarrer les activités 100, 200, 300
- Attendre la fin des activités 100, 200, 300
- Démarrer l'activité 400
- Attendre la fin de l'activité 400

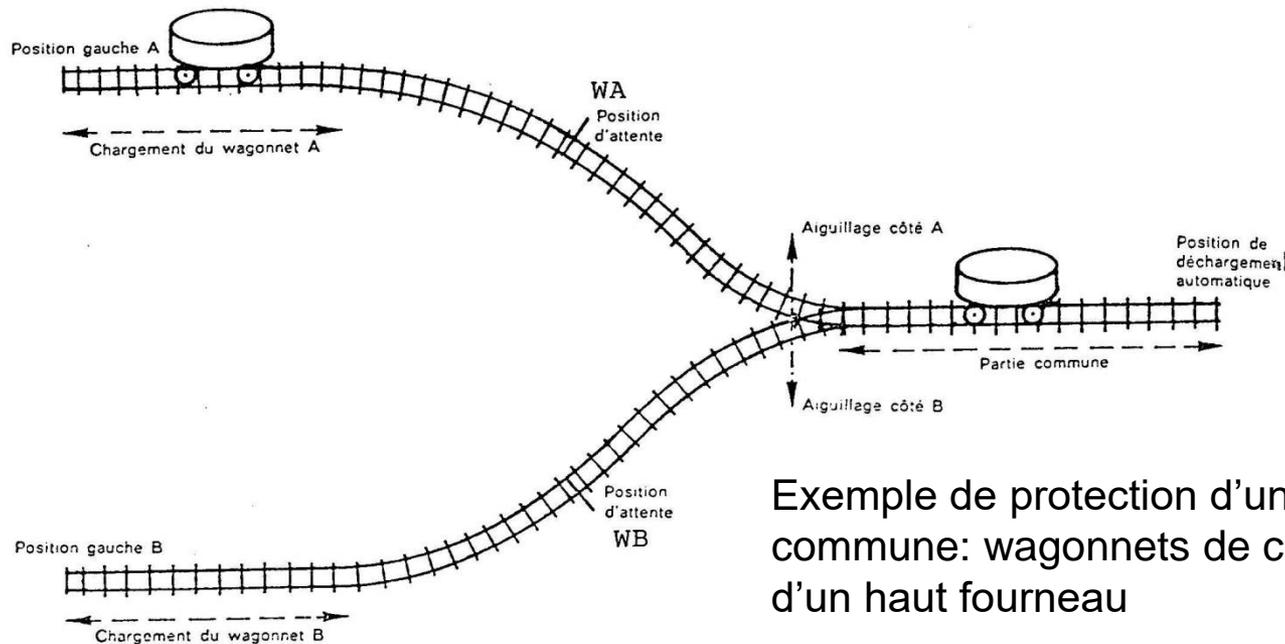
Synchronisation verticale

GRAFCET

- METHODOLOGIE DE GESTION D'UN PROJET
- Structuration
 - Approche top down
 - Utilisation de sous grafkets
- Synchronisation : gestion du multitâche
 - Le mode de représentation GRAFCET autorise le lancement d'activités en parallèle et se déroulant de manière indépendante
 - Nécessité de resynchroniser ces activités
- **Protection des ressources communes**
 - **Eviter les accès simultanés**
- Transposition du GRAFCET en langage automate

GRAF CET: Protection des ressources communes

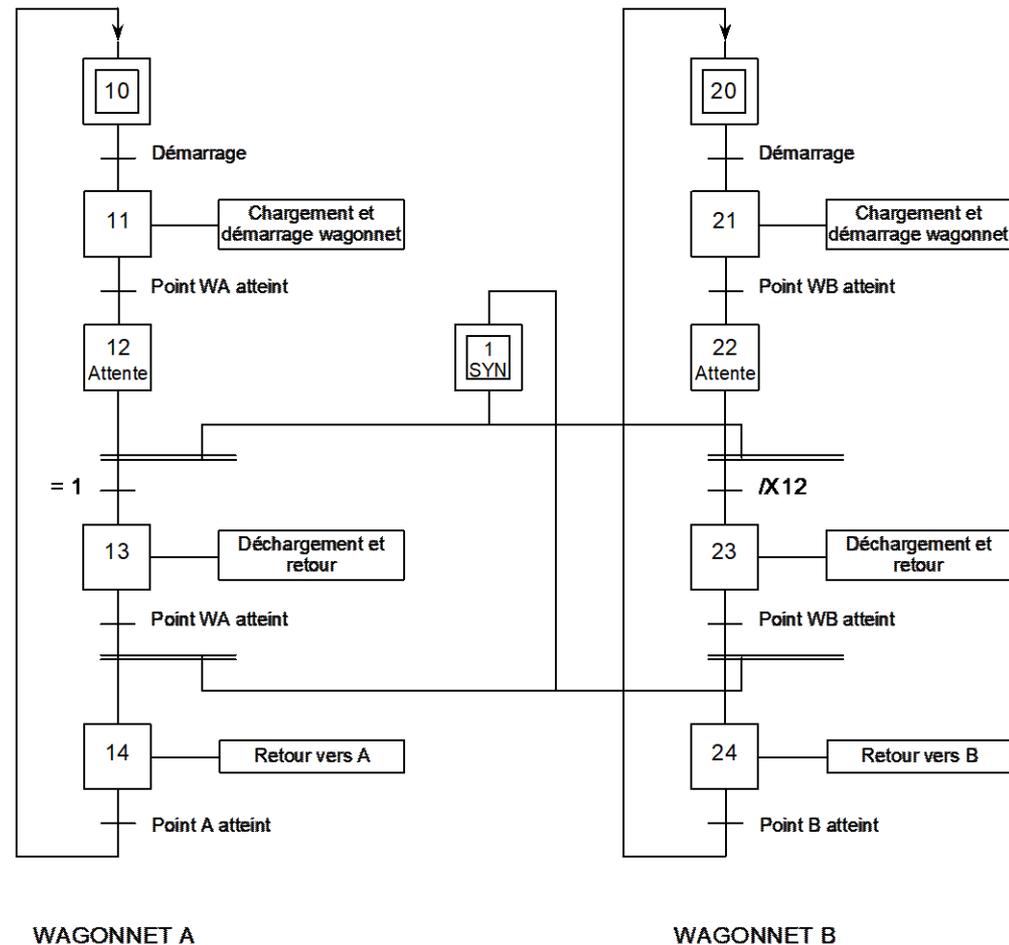
- La protection des ressources communes se pose typiquement lorsque l'on traite des activités se déroulant en parallèle.
- Les ressources communes doivent être protégées



Exemple de protection d'une ressource commune: wagonnets de chargement d'un haut fourneau

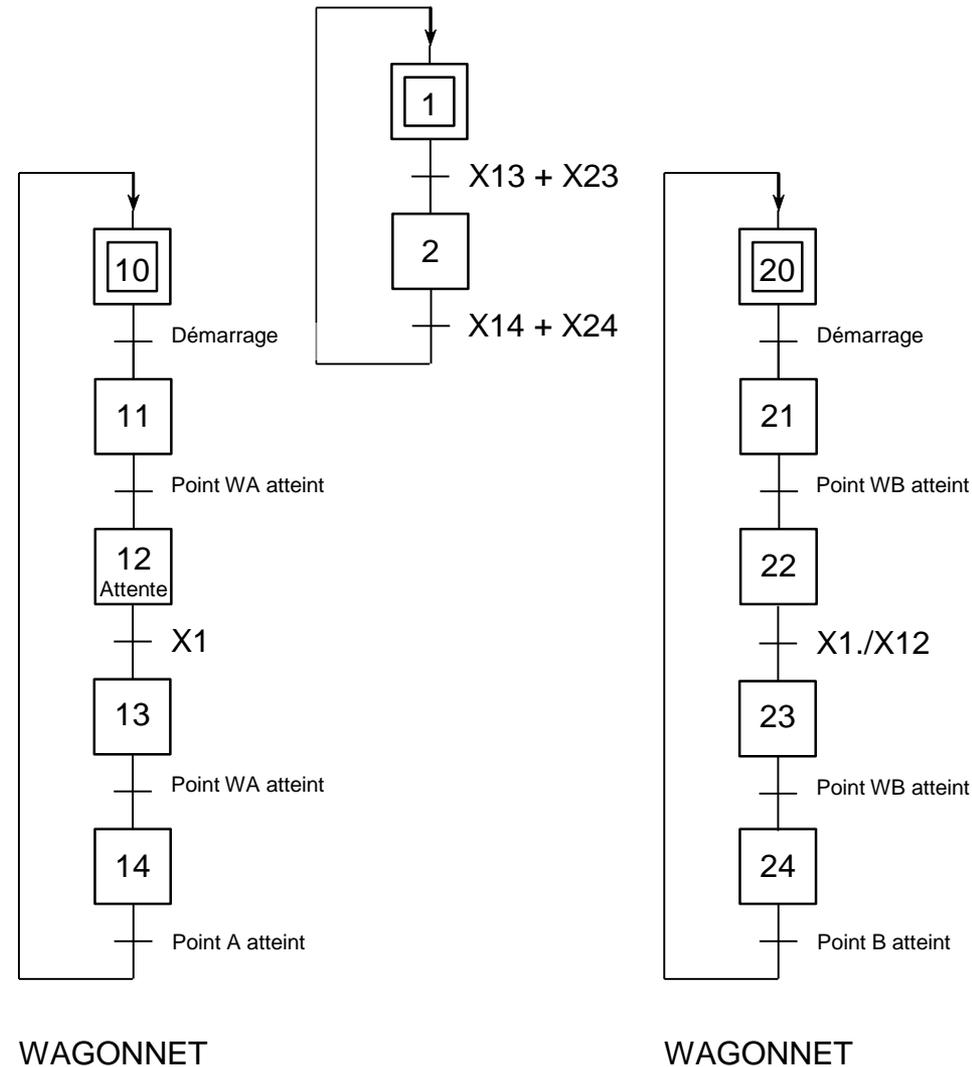
GRAFCET: Protection des ressources communes

- Le GRAFCET permet de gérer ces conflits de manière élégante
- Utilisation d'une étape de verrouillage
- L'état libre ou occupé de la ressource commune est représenté par la variable d'étape X1
- Pour éviter le cas où les deux wagons se présentent simultanément en WA et WB, on a la réceptivité /X12, ce qui donne priorité au wagonnet A.



GRAFNET: Protection des ressources communes

- Mécanisme de protection de la ressource commune par l'utilisation d'un graphe de synchronisation
- Remarque : étape 1 joue le rôle de sémaphore en programmation temps réel



GRAFCET

- METHODOLOGIE DE GESTION D'UN PROJET
- Structuration
 - Approche top down
 - Utilisation de sous grafkets
- Synchronisation : gestion du multitâche
 - Le mode de représentation GRAFCET autorise le lancement d'activités en parallèle et se déroulant de manière indépendante
 - Nécessité de resynchroniser ces activités
- Protection des ressources communes
 - Eviter les accès simultanés
- **Transposition du GRAFCET en langage automate**

GRAFCET: TRANSPOSITION EN LANGAGE AUTOMATE

Transposition en langage automate

- Types de méthodes:

- Bits d'étape:

- L'activité d'une étape est représentée par l'état vrai ou faux d'une variable binaire interne
 - Plusieurs variantes qui peuvent être programmées en IL, Ladder Diagram, etc.

- Pointeur de phase;

- L'activité est représentée par la valeur d'un mot
 - Valeur du mot = numéro de l'étape active
 - Affecter un mot aux branches susceptibles d'évoluer en parallèle
 - Travail sur des mots \Rightarrow langages adéquats ST

Transposition en langage automate

■ Critères de choix

- Facilité d'édition et lisibilité du programme
 - Adéquation entre méthode et langage utilisé
 - Exemple:
 - Structured Text le pointeur de phase est meilleur
 - IL: bit d'étape
- Facilité de mise au point
 - Forçage des variables en phase de mise au point
 - Intérêt et sécurité du pointeur de phase
- Facilité de diagnostic
 - Visualisation dynamique du pointeur de phase
- Performances:
 - Dépend du type d'automate et du nombre d'étapes du GRAFCET
 - Pas de règle générale

Transposition en langage automate

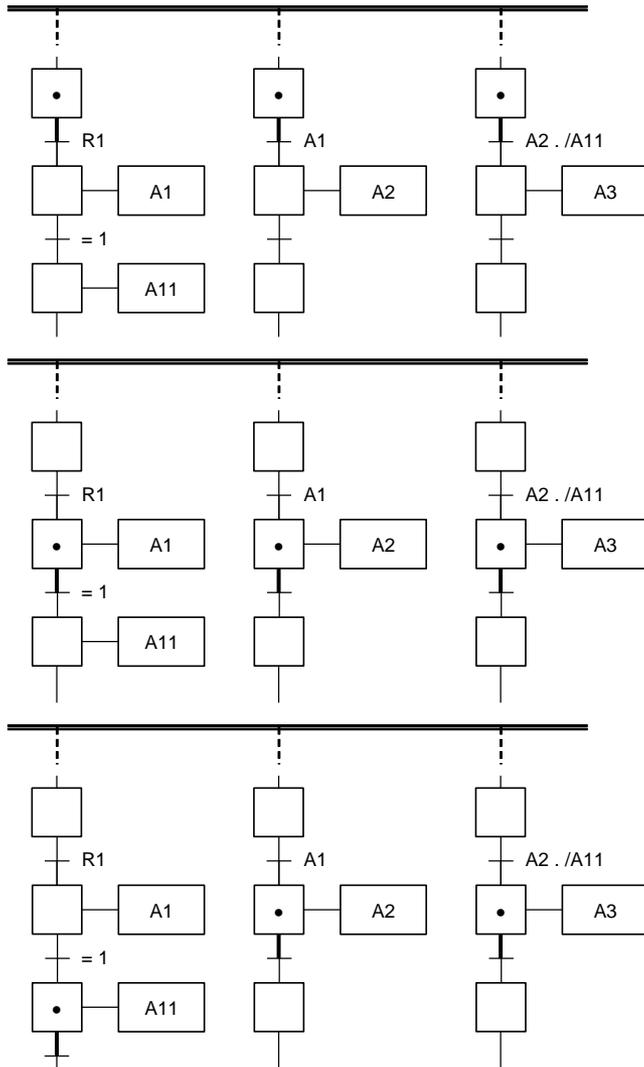
- Difficultés potentielles
 - Cause: les instructions décrivant le GRAFCET ne sont pas exécutées de manière simultanée, mais de manière séquentielle.
- **Durée d'activation minimale d'une étape doit être au moins égale à un temps de cycle**
 - Les sorties sont calculées durant le cycle, mais elles ne sont inscrites dans la table des images et envoyées vers les interfaces qu'à la fin du cycle
 - Si étape activée et désactivée au cours d'un même cycle: pas d'effet!
 - Règle: **les étapes doivent rester actives au moins un temps de cycle d'automate**
- Conditions d'activation de course potentiellement liées à l'ordre de scrutation des GRAFCETs.

Transposition en langage automate

- Problème de parallélisme d'exécution
 - Pas de solution radicale et universelle
 - L'utilisateur doit détecter le problème et organiser la programmation pour contrer le problème
 - Etudier la logique (si elle est donnée) avec laquelle le programme exécute le programme (par exemple en logigramme de droite à gauche et de haut en bas)

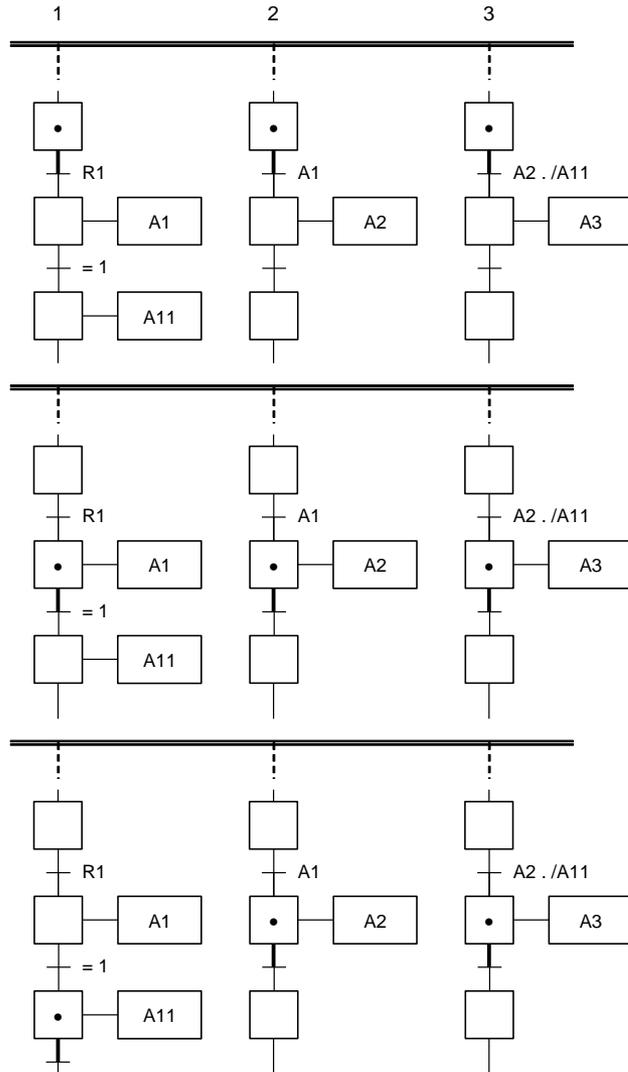
- Problème de synchronisation
 - Problème lié à l'échange de variables d'étape pour synchroniser, pour le parallélisme et pour la protection des ressources communes
 - Dans ce cas, la difficulté est contournée de la manière suivante: lorsqu'un GRAFCET prend le contrôle de la ressource, il fait évoluer dans le même temps le GRAFCET de synchronisation
 - On duplique donc les évolutions du GRAFCET de contrôle

Problème des branches parallèles interactives



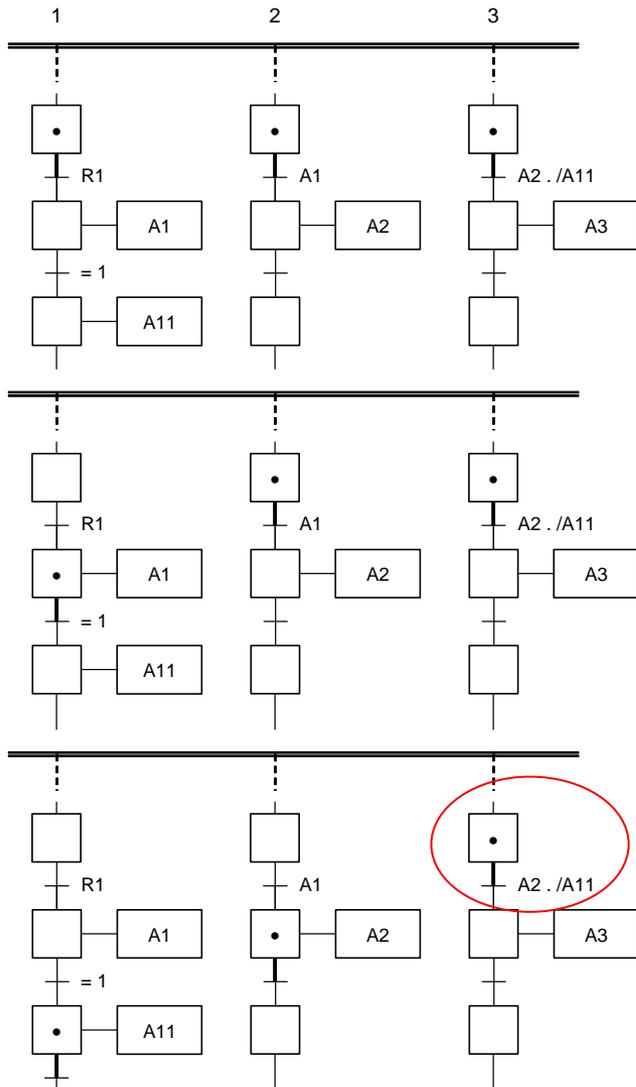
Cas idéal: évolution simultanée

Problème des branches parallèles interactives



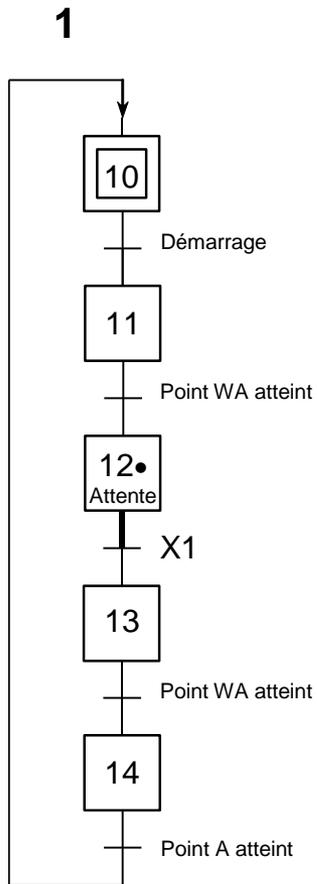
Scrutation dans l'ordre 1-2-3:
On atteint l'état stable correct après
deux cycles

Problème des branches parallèles interactives

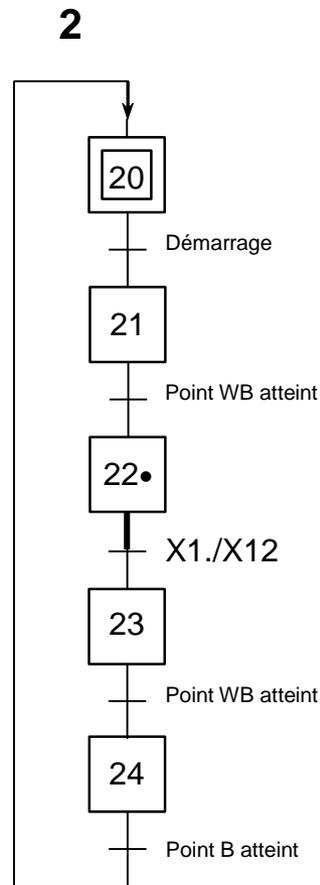


Scrutation dans l'ordre 3-2-1:
On atteint l'état stable erroné après
deux cycles

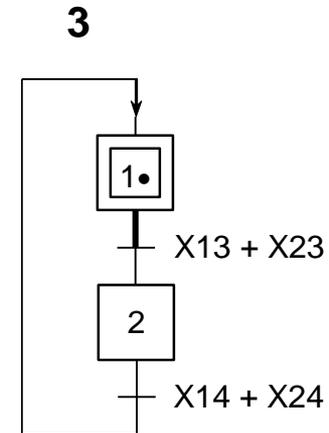
Problème de synchronisation



WAGONNET

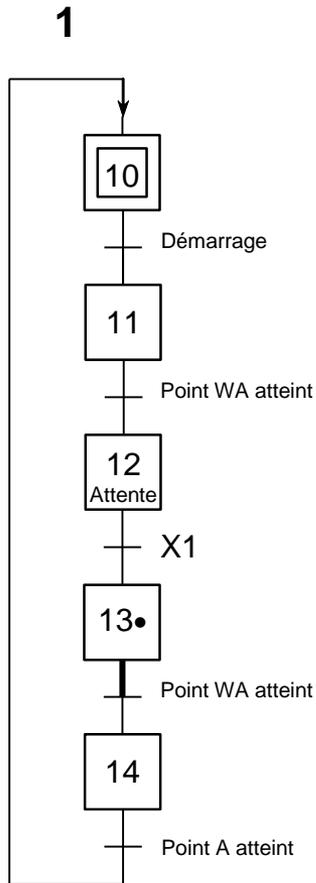


WAGONNET

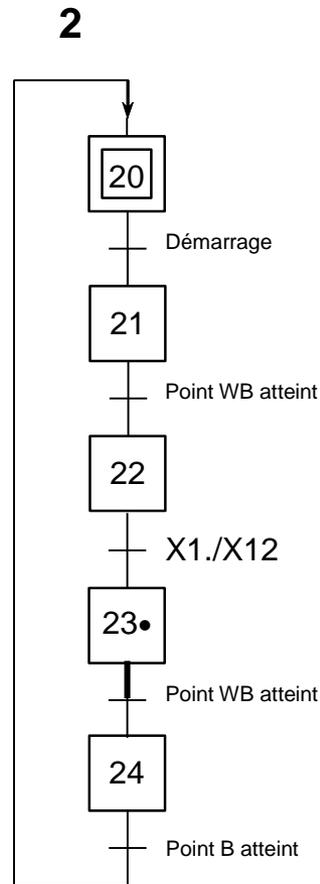


Etat initial

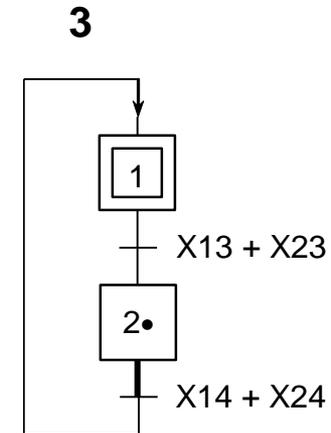
Problème de synchronisation



WAGONNET



WAGONNET



Etat final:
Evolution fautive dans le cas
d'une scrutation 1 – 2 -3

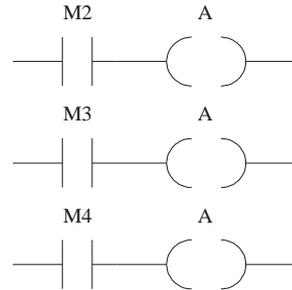
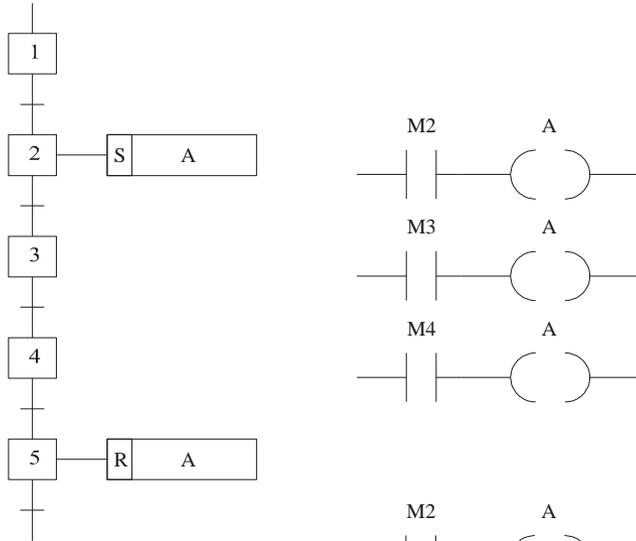
TRANSPOSITION DU GRAFCET EN LANGAGE

Organisation des programmes

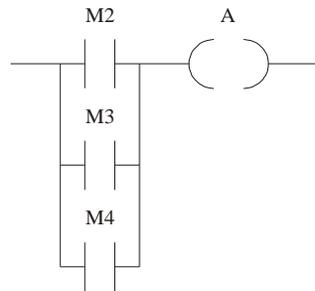
- Partie combinatoire
 - Séparer la partie combinatoire de la partie séquentielle
 - Mettre la partie combinatoire en avant

- Partie séquentielle
 - Séparer la partie du programme relative à l'évolution du GRAFCET et la partie calcul des actions
 - Les actions
 - Calcul des actions en fin de programme
 - Trier le calcul des actions en une seule équation
 - Facilité de la mise au point
 - Traitement plus facile des arrêts d'urgence

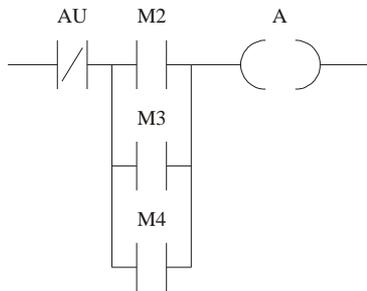
Organisation des programmes: actions



a.



b.



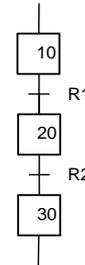
c.

Intérêt de regrouper les équations relatives aux actions

Méthodes de transpositions

- **METHODE 1: CALCUL PRÉALABLE DES CONDITIONS DE TRANSITION**
 - Les conditions de transition pour toutes les étapes sont calculées en début de programme
 - Les bits internes de résultats sont ensuite utilisés en tant que réceptivités dans la description de l'évolution du GRAFCET
 - Si une étape était activée au cours du cycle, on est certain qu'elle ne peut pas être désactivée dans un même cycle d'automate puisque la réceptivité + validation est nécessairement faux
 - Les actions sont regroupées à la fin du programme

- Exemple de GRAFCET



- Calcul des conditions de transition

```
LD      M10
AND     R1
ST      M11
LD      M20
AND     R2
ST      M21
.
```

condition de transition 10 - 20

Si l'étape 10 est active
et que la réceptivité R1 est vraie
activer la condition de transition M11
condition de transition 20 - 30
Si l'étape 20 est active
et que la réceptivité R2 est vraie
activer la condition de transition M21

- Calcul de l'évolution

```
LD      M10
AND     M11
S       M20
R       M10
```

transition 10, 20

Si l'étape 10 est active
et que la condition de transition est vraie
alors activer l'étape 20
et désactiver l'étape 10

```
LD      M20
AND     M21
S       M30
R       M20
.
```

transition 20, 30

Si l'étape 20 est active
et que la condition de transition est vraie
alors activer l'étape 30
et désactiver l'étape 20

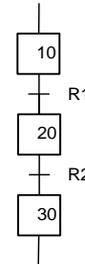
- Calcul des actions

```
.
```

Méthodes de transpositions

- Calcul préalable des conditions de transition
 - Lourd point de vue calcul
 - Forçage facile (un bit à forcer)
 - Purement logique (pas d'opérations arithmétiques ni de sauts)
 - Assure un temps de cycle au moins par étape
 - Assure la condition pour les actions « impulsionnelles »

- Exemple de GRAFCET



- Calcul des conditions de transition

```
LD      M10
AND     R1
ST      M11
LD      M20
AND     R2
ST      M21
.
```

condition de transition 10 - 20

Si l'étape 10 est active
et que la réceptivité R1 est vraie
activer la condition de transition M11
condition de transition 20 - 30
Si l'étape 20 est active
et que la réceptivité R2 est vraie
activer la condition de transition M21

- Calcul de l'évolution

```
LD      M10
AND     M11
S       M20
R       M10
```

transition 10, 20

Si l'étape 10 est active
et que la condition de transition est vraie
alors activer l'étape 20
et désactiver l'étape 10

```
LD      M20
AND     M21
S       M30
R       M20
.
```

transition 20, 30

Si l'étape 20 est active
et que la condition de transition est vraie
alors activer l'étape 30
et désactiver l'étape 20

- Calcul des actions

```
.
```

Méthodes de transpositions

■ METHODE 2: UTILISATION DES SAUTS CONDITIONNELS

- Si le langage comporte des sauts conditionnels, on peut les utiliser pour éviter des calculs inutiles
- On saute d'étape en étape jusqu'à tomber sur l'étape active
- On calcule la réceptivité pour l'étape active et si elle est vraie on fait évoluer le GRAFCET
- On saute à la fin du programme
- Les actions sont regroupées à la fin du programme

ET10 :	LDN	M10	transition 10 - 20
	JMPC	ET20	Si l'étape 10 n'est pas active
	LD	R1	sauter à l'étape suivante
	S	M20	Sinon tester la réceptivité et, si elle est vraie
	R	M10	activer l'étape 20
	JMP	ACTION	désactiver l'étape 10
			sauter au calcul des actions
ET20 :	LDN	M20	transition 20 - 30
	JMPC	ET30	Si l'étape 20 n'est pas active
	LD	R2	sauter à l'étape suivante
	S	M30	Sinon tester la réceptivité et, si elle est vraie
	R	M20	activer l'étape 30
	JMP	ACTION	désactiver l'étape 20
	·		sauter au calcul des actions
	·		
	·		
ACTION :			calcul des actions
	·		
	·		

Méthodes de transpositions

- Utilisation des sauts conditionnels
 - Plus difficile pour le forçage des variables car il faut recalculer R1
 - On peut envisager un calcul préalable des réceptivités avec un bit interne de mémorisation, mais on perd une partie du bénéfice des sauts.

ET10 :	LDN	M10	transition 10 - 20
	JMPC	ET20	Si l'étape 10 n'est pas active
	LD	R1	sauter à l'étape suivante
	S	M20	Sinon tester la réceptivité et, si elle est vraie
	R	M10	activer l'étape 20
	JMP	ACTION	désactiver l'étape 10
			sauter au calcul des actions
ET20 :	LDN	M20	transition 20 - 30
	JMPC	ET30	Si l'étape 20 n'est pas active
	LD	R2	sauter à l'étape suivante
	S	M30	Sinon tester la réceptivité et, si elle est vraie
	R	M20	activer l'étape 30
	JMP	ACTION	désactiver l'étape 20
	•		sauter au calcul des actions
	•		
	•		
ACTION :			calcul des actions
	•		
	•		
	•		

Méthodes de transpositions

■ METHODE 3: UTILISATION D'UN POINTEUR DE PHASE

- Pour les automates disposant de sauts conditionnels et d'opérations sur mots
- Si le GRAFCET a des branches parallèles, on le replace par autant de GRAFCET simples synchronisés
- A chaque GRAFCET on associe un mot de données appelé **pointeur de phase** dont la valeur est égale au numéro de l'étape active.
- **Condition de transition:** comparaison avec le numéro de l'étape et la valeur du pointeur de phase
- **Evolution:** le pointeur prend la valeur de la nouvelle étape active

a. Programmation en langage IL

Pointeur de phase MW1

```
ET10 : LD 10
      NE MW1
      JMPC ET20
      LD R1
      JMPCN ACTION
      LD 20
      ST MW1
      JMP ACTION

ET20 : LD 20
      NE MW1
      JMPC ET30
      LD R2
      JMPCN ET20
      LD 30
      ST MW1
      JMP ACTION
      .
      .
      .

ACTION :
      .
      .
      .
```

transition 10 - 20
Si le pointeur de phase n'est pas égal à 10
sauter à l'étape suivante
sinon, tester la réceptivité
sauter au calcul des actions si elle est fausse
sinon mettre 20 dans le pointeur de phase
sauter au calcul des actions

transition 20 - 30
Si le pointeur de phase n'est pas égal à 20
sauter à l'étape suivante
sinon, tester la réceptivité
sauter au calcul des actions si elle est fausse
sinon mettre 30 dans le pointeur de phase
sauter au calcul des actions

b. Programmation en langage ST

```
! IF [MW1=10] . R1
      THEN 20 → MW1 ; JUMP ACTION

! IF [MW1=20] . R2
      THEN 30 → MW1 ; JUMP ACTION
      .
      .
      .

ACTION :
      .
      .
      .
```

transition 10 - 20

transition 20 - 30

Méthodes de transpositions

■ Utilisation d'un pointeur de phase

□ Désavantage

- Méthode plus gourmande en temps de calcul car opérations sur mots

□ Avantages:

- Garantit de manière intrinsèque qu'il n'y a qu'une seule étape active
- Facilité du forçage
- Facilité de la visualisation
- Lisibilité

a. Programmation en langage IL

Pointeur de phase MW1

```
ET10 : LD 10
        NE MW1
        JMP  ET20
        LD R1
        JMPCN ACTION
        LD 20
        ST MW1
        JMP ACTION
```

transition 10 - 20
Si le pointeur de phase n'est pas égal à 10
sauter à l'étape suivante
sinon, tester la réceptivité
sauter au calcul des actions si elle est fausse
sinon mettre 20 dans le pointeur de phase
sauter au calcul des actions

```
ET20 : LD 20
        NE MW1
        JMP  ET30
        LD R2
        JMPCN ET20
        LD 30
        ST MW1
        JMP ACTION
        .
        .
        .
```

transition 20 - 30
Si le pointeur de phase n'est pas égal à 20
sauter à l'étape suivante
sinon, tester la réceptivité
sauter au calcul des actions si elle est fausse
sinon mettre 30 dans le pointeur de phase
sauter au calcul des actions

ACTION : calcul des actions
:
:

b. Programmation en langage ST

```
! IF [MW1=10] . R1
    THEN 20 → MW1 ; JUMP ACTION
```

transition 10 - 20

```
! IF [MW1=20] . R2
    THEN 30 → MW1 ; JUMP ACTION
```

transition 20 - 30

ACTION :
:
: