

---

## Objet de ce chapitre

Ce chapitre décrit le langage séquentiel SFC conforme à la norme CEI 61131.1.

## Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
13.1	Généralités concernant le diagramme fonctionnel en séquence (SFC)	400
13.2	Etape et macro-étape	405
13.3	Action et section d'action	415
13.4	Transition et section de transition	422
13.5	Saut	427
13.6	Liaison	428
13.7	Divergences et convergences	429
13.8	Objet texte	432
13.9	Jeton unique	433
13.10	Jetons multiples	444

## 13.1 Généralités concernant le diagramme fonctionnel en séquence (SFC)

---

### Objet de ce sous-chapitre

Ce sous-chapitre vous donne un aperçu général du diagramme fonctionnel en séquence SFC.

### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Informations générales sur le diagramme fonctionnel en séquence SFC	401
Règles de liaison	404

## Informations générales sur le diagramme fonctionnel en séquence SFC

### Présentation

Cette section décrit le langage séquentiel SFC (Diagramme fonctionnel en séquence), conforme à la norme CEI 61131-3.

### Structure d'un diagramme fonctionnel en séquence

Un diagramme fonctionnel en séquence conforme à CEI se compose dans Unity Pro de sections SFC (niveau supérieur), de sections de transition et de sections d'action.

Ces sections SFC ne sont admises que dans la tâche maître du projet. Dans d'autres tâches ou DFB, les sections SFC ne peuvent pas être utilisées.

Chaque section SFC contient exactement un réseau SFC (séquence) dans le jeton unique.

Les jetons multiples d'une section SFC peuvent contenir un ou plusieurs réseaux SFC indépendants les uns des autres.

### Objets

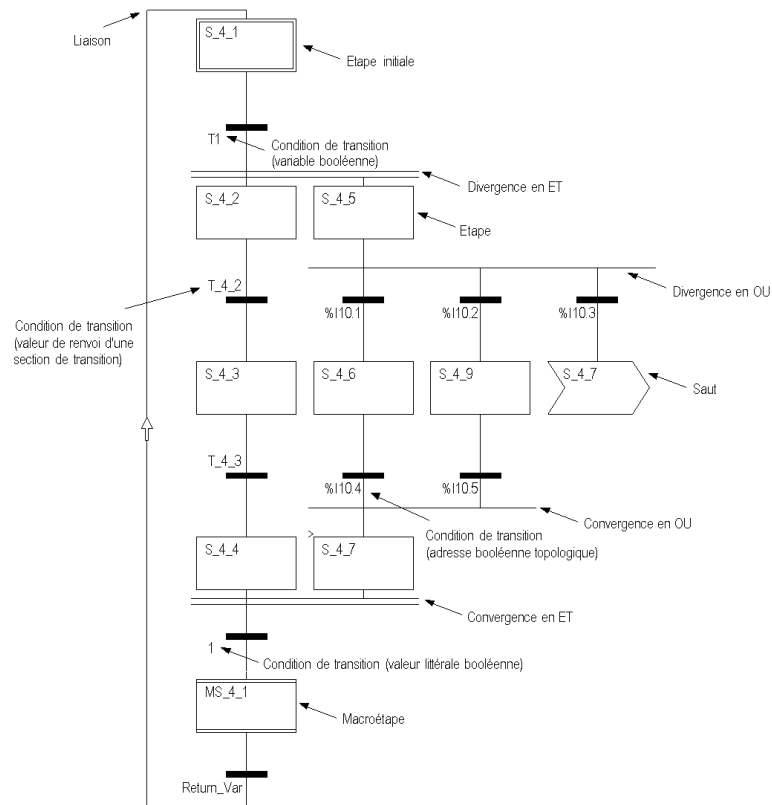
Une section SFC contient les objets de création de programme suivants :

- étape (*voir page 406*)
- macroétape (séquence de sous-étapes imbriquées) (*voir page 411*)
- transition (condition de transition) (*voir page 423*)
- saut (*voir page 427*)
- liaison (*voir page 428*)
- divergence en OU (*voir page 430*)
- convergence en OU (*voir page 430*)
- divergence en ET (*voir page 431*)
- convergence en ET (*voir page 431*)

La logique de la section peut être commentée par des objets texte (*voir Objet texte, page 432*).

## Représentation d'une section SFC

Représentation :



### Structure d'une section SFC

Une section SFC est une "machine d'états", c.-à-d. que l'état est déterminé par les étapes actives et les transitions renvoient le comportement de commutation/modification entre les états. Les étapes et transitions sont reliées les unes aux autres par des liaisons dirigées. Deux étapes consécutives ne peuvent jamais être directement reliées ; elles sont toujours séparées par une transition. Les évolutions des états actifs de signaux se déroulent le long des liaisons dirigées, et sont déclenchées par la commutation d'une transition. Le déroulement d'une séquence va dans le sens des liaisons dirigées et se déroule de la partie inférieure de l'étape précédente à la partie supérieure de l'étape suivante. Les divergences sont traitées de gauche à droite.

Chaque étape peut compter zéro ou plusieurs actions. A chaque transition est associée une condition de transition.

La dernière transition de la séquence est toujours reliée à une autre étape de la séquence (par une liaison graphique ou un symbole de "saut") de manière à obtenir une boucle fermée. Les séquences d'étapes se déroulent donc de façon cyclique.

### SFCCHART\_STATE Variable

A la création d'une section SFC, une variable du type de données SFCCHART\_STATE lui est automatiquement affectée. La variable ainsi créée porte toujours le nom de la section SFC correspondante.

Cette variable sert à affecter les blocs de commande SFC à la section SFC à commander.

### Règle de jetons

Le comportement d'un réseau SFC dépend largement du nombre de jetons choisis, c.-à-d. du nombre d'étapes actives.

Un comportement univoque est possible en utilisant un seul jeton (single token). Les divergences en ET comportant un jeton actif (étape) par branche sont considérées comme des jetons uniques. Ceci correspond à une séquence d'étapes selon la norme CEI 61131-3.

Une séquence d'étapes comportant un maximum d'étapes actives (jetons multiples) définies par l'utilisateur augmente le niveau de liberté. Les limitations relatives à l'obligation d'unicité et du non-blocage sont à cet effet levées et doivent être assurées par l'utilisateur. Les séquences d'étape à jetons multiples ne sont pas conformes à la norme CEI 61131-3.

### Taille de la section

- Une section SFC se compose d'une fenêtre comportant une seule page.
- Pour des raisons de performance, il est recommandé de créer moins de 100 sections SFC dans un projet (les sections macro ne sont pas comptabilisées).
- Cette fenêtre comporte une grille logique de 200 lignes et de 32 colonnes.
- Les étapes, les transitions et les sauts requièrent tous une cellule.
- Les divergences et convergences ne nécessitent pas de cellule propre, mais sont insérées dans la cellule correspondante de l'étape ou de la transition.
- Chaque section SFC (avec toutes ses macrosections) peut contenir jusqu'à 1024 étapes.
- Il est possible d'activer jusqu'à 100 étapes (jetons multiples) par section SFC (avec toutes ses macrosections).
- Il est possible de placer manuellement jusqu'à 64 étapes simultanément par section SFC (jetons multiples).
- 20 actions, au maximum, peuvent être affectées à chaque étape SFC.
- La possibilité d'imbrication des macros, c'est-à-dire "macroétape dans macroétape", s'élève à huit niveaux.

### Conformité CEI

Pour la description de la conformité CEI du langage SFC, voir Conformité CEI (voir page 657).

## Règles de liaison

### Règles de liaison

Ce tableau indique les connexions que vous pouvez établir entre les sorties d'objet et les entrées d'objet.

Depuis la sortie de l'objet	Vers l'entrée de l'objet
Etape	Transition
	Divergence en OU
	Convergence en ET
Transition	Etape
	Saut
	Divergence en ET
	Convergence en OU
Divergence en OU	Transition
Convergence en OU	Etape
	Saut
	Divergence en ET
	Convergence en OU
Divergence en ET	Etape
	Saut
	Convergence en OU (uniquement pour les jetons multiples ( <i>voir page 444</i> ))
Convergence en ET	Transition
	Divergence en OU (uniquement pour jetons multiples ( <i>voir page 444</i> ))
	Convergence en OU

---

## 13.2 Etape et macro-étape

---

### Objet de ce sous-chapitre

Ce sous-chapitre décrit les objets étape et macro-étape du diagramme fonctionnel en séquence (SFC).

### Contenu de ce sous-chapitre

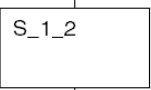
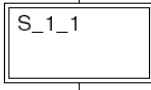
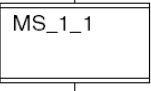
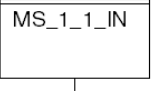
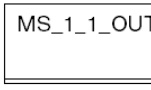
Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Etape	406
Macro-étapes et macro-sections	411

## Étape

### Types d'étape

Les étapes se composent des types suivants :

Type	Représentation	Description
Étape "normale"		Une étape devient active lorsque l'étape précédente devient inactive (un temps de retard éventuellement défini doit s'être écoulé) et la transition située en amont est vraie. Une étape devient généralement inactive lorsque le temps de retard éventuellement défini s'est écoulé et que la transition située en aval est vraie. Pour les convergences en ET, toutes les étapes précédentes doivent être vraies. Chaque étape compte zéro ou plusieurs actions. Les étapes sans action sont considérées comme des étapes d'attente.
Étape initiale		L'état initial d'une séquence est caractérisé par l'étape initiale. A l'issue de l'initialisation du projet ou de la séquence, l'étape initiale est active. Généralement, aucune action n'est affectée aux étapes initiales. Pour les jetons uniques (conformes à la norme CEI 61131-3), seule une étape initiale est admise par séquence. Pour les jetons multiples un nombre d'étapes initiales pouvant être défini (de 0 à 100) est possible.
Macroétape		voir <i>Macroétape, page 411</i>
Étape d'entrée		voir <i>Étape d'entrée, page 411</i>
Étape de sortie		voir <i>Étape de sortie, page 412</i>



## Noms d'étape

A chaque création d'étape, un numéro de proposition lui est affecté. La structure du numéro proposé est la suivante :  $S_{i_j}$ , où  $i$  représente le numéro reel (interne) de la section et  $j$  est le numéro de l'étape actuelle (interne) dans la section.

Vous pouvez changer ces numéros de proposition pour avoir une meilleure vue d'ensemble. Les noms d'étape (32 caractères maximum) doivent être uniques dans l'ensemble du projet, c-à-d. qu'il ne doit pas exister d'autres étapes, variables, sections etc., ayant le même nom. Aucune distinction n'est faite entre l'écriture en majuscules et en minuscules. Le nom d'étape doit répondre aux conventions sur les noms.

## Temps d'étape

Un temps minimum et maximum de contrôle ainsi qu'un temps de retard peuvent être affectés à chaque étape.

- **Temps de contrôle minimum**

Le temps minimum de contrôle indique la durée minimale pendant laquelle l'étape doit être active. Si l'étape devient inactive avant que ce délai ne soit écoulé, un message d'erreur s'affiche. En mode d'animation, l'erreur est de plus signalée par un changement de couleur (jaune) de l'objet d'étape.

Si vous n'indiquez pas de temps minimum de contrôle ou que vous indiquez un temps minimum nul, un contrôle de l'étape n'est pas exécuté.

L'état d'erreur est conservé jusqu'à ce que l'étape redevienne active.

- **Temps maximum de contrôle**

Le temps maximum de contrôle indique la durée maximale pendant laquelle l'étape doit être normalement active. Si l'étape est toujours active lorsque ce délai est écoulé, un message d'erreur apparaît. En mode d'animation, l'erreur est de plus signalée par un changement de couleur (rose) de l'objet d'étape.

Si vous n'indiquez pas de temps maximum de contrôle ou que vous indiquez un temps maximum nul, un contrôle de l'étape n'est pas exécuté.

L'état d'erreur est conservé jusqu'à ce que l'étape devienne inactive.

- **Temps de retard**

Le temps de retard (palier de l'étape) indique la durée minimale pendant laquelle l'étape doit être active.

**NOTE :** Les durées indiquées ne sont valables que pour l'étape, pas pour les actions qui y sont associées. Pour celles-ci, il est possible de définir des temps propres.

### Définition des temps d'étapes

Lors de la définition/détermination des temps, veuillez tenir compte de la formule suivante :

Temps de retard < temps minimum de contrôle < temps maximum de contrôle

On distingue 2 possibilités d'affectation des valeurs définies/déterminées à une étape :

- saisie sous forme de libellé de durée
- utilisation de la structure de données `SFCSTEP_TIMES`

**Variable SFCSTEP\_TIMES**

Vous pouvez affecter à chaque étape une variable du type de données SFCSTEP\_TIMES. Les éléments de cette structure de données sont accessibles en lecture et en écriture (read/write).

La structure de données est traitée comme toute autre structure de données, c.-à-d. qu'elle peut être utilisée dans des déclarations de variables et il est donc possible d'y accéder dans son ensemble (p. ex. en tant que paramètre FFB).

Description de la structure de données :

Nom de l'élément	Type	Description
"VarName".delay	TIME	Temps de retard
"VarName".min	TIME	Temps de contrôle minimum
"VarName".max	TIME	Temps maximum de contrôle

**Variable SFCSTEP\_STATE**

A chaque étape est affectée de façon implicite une variable du type de données SFCSTEP\_STATE. Cette variable d'étape porte le nom de l'étape à laquelle elle est affectée. Les éléments de cette structure de données ne sont accessibles qu'en lecture (read-only).

Vous pouvez voir les variables SFCSTEP\_STATE dans l' **Editeur de données**. Le **Commentaire** pour une variable SFCSTEP\_STATE est le commentaire entré comme propriété de l'étape elle-même. Consultez la section "Définition des propriétés d'étapes" (voir *Unity Pro, Modes de marche*, ) dans le *manuel des modes de fonctionnement Unity Pro*.

La structure de données ne peut pas être utilisée dans des déclarations de variables. C'est pourquoi il est impossible d'accéder à la structure de données dans son ensemble (p. ex. sous forme de paramètre FFB).

## Description de la structure de données :

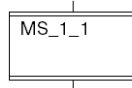
Nom de l'élément	Type	Description
"Nom d'étape" t	TIME	Temps d'activation actuel dans l'étape. Lorsque l'étape est désactivée, la valeur de cet élément est maintenue jusqu'à la prochaine activation de l'étape.
"Nom d'étape" .x	BOOL	1: Etape active 0: Etape inactive
"Nom d'étape".tminErr	BOOL	Cet élément est une extension de la norme CEI 61131-3. 1: dépassement par valeur inférieure du temps minimum de contrôle 0: pas de dépassement par valeur inférieure du temps minimum de contrôle Dans les cas suivants, l'élément est automatiquement réinitialisé : <ul style="list-style-type: none"> <li>● lorsque l'étape est réactivée,</li> <li>● lorsque la commande de séquence est réinitialisée,</li> <li>● Si le bouton de commande <b>RAZ erreur durée</b> est activé</li> </ul>
"Nom d'étape".tmaxErr	BOOL	Cet élément est une extension de la norme CEI 61131-3. 1: Dépassement du temps de contrôle maximum 0: Pas de dépassement du temps maximum de contrôle Dans les cas suivants, l'élément est automatiquement réinitialisé : <ul style="list-style-type: none"> <li>● à la sortie de l'étape,</li> <li>● lorsque la commande de séquence est réinitialisée,</li> <li>● Si le bouton de commande <b>RAZ erreur durée</b> est activé</li> </ul>

## Macro-étapes et macro-sections

### Macroétape

Les macro-étapes servent à appeler une macro-section et ainsi à établir une structure hiérarchique des commandes d'enchaînement.

Représentation d'une macroétape :



Les macroétapes ont les caractéristiques suivantes :

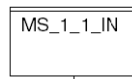
- elles se positionnent en sections de "commande de séquence" et en macro-sections,
- le nombre de macroétapes n'est pas limité,
- le degré d'imbrication des macroétapes l'une dans l'autre est de 8 niveaux,
- à chaque macroétape est affectée de manière implicite une variable du type de données `SFCSTEP_STATE`, voir *Variable SFCSTEP\_STATE, page 409*,
- à chaque macroétape peut être affectée une variable du type de données `SFCSTEP_TIMES`, voir *Variable SFCSTEP\_TIMES, page 409*,
- AUCUNE action ne peut être affectée aux macroétapes,
- Chaque macroétape peut être remplacée par la séquence contenue dans la macro-section correspondante.

Les macroétapes sont une extension de la norme CEI 61131-3 et doivent être activées de manière explicite.

### Etape d'entrée

Chaque macro-section commence par une étape d'entrée.

Représentation d'une étape d'entrée :



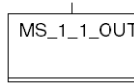
Les étapes d'entrée ont les caractéristiques suivantes :

- les étapes d'entrée sont automatiquement placées dans des sections de macro par l'éditeur SFC,
- seule une étape d'entrée est placée par macro-section,
- une étape d'entrée ne peut pas être supprimée, copiée ni insérée manuellement,
- à chaque étape d'entrée est affectée de manière implicite une variable du type de données `SFCSTEP_STATE`, voir *Variable SFCSTEP\_STATE, page 409*,
- à chaque étape d'entrée peut être affectée une variable du type de données `SFCSTEP_TIMES`, voir *Variable SFCSTEP\_TIMES, page 409*,
- des actions peuvent être affectées à des étapes d'entrée.

## Etape de sortie

Chaque macro-section se termine par une étape de sortie.

Représentation d'une étape de sortie :



Une étape de sortie a les caractéristiques suivantes :

- les étapes de sortie sont automatiquement placées dans des sections de macro par l'éditeur SFC,
- seule une étape de sortie est placée par macro-section,
- une étape de sortie ne peut pas être supprimée, copiée ni insérée manuellement,
- AUCUNE action ne peut être affectée aux étapes de sortie,
- les étapes de sortie ne peuvent se voir affecter qu'un temps de retard, l'affectation de temps de contrôle n'est pas possible, voir *Temps d'étape*, page 407.

## Macro-section

Une macro-section se compose d'une séquence unique, laquelle dispose en principe des mêmes éléments que les sections de "commande de séquence" (p.ex. étapes, étape(s) initiale(s), macro-étapes, transitions, divergences, convergences etc.).

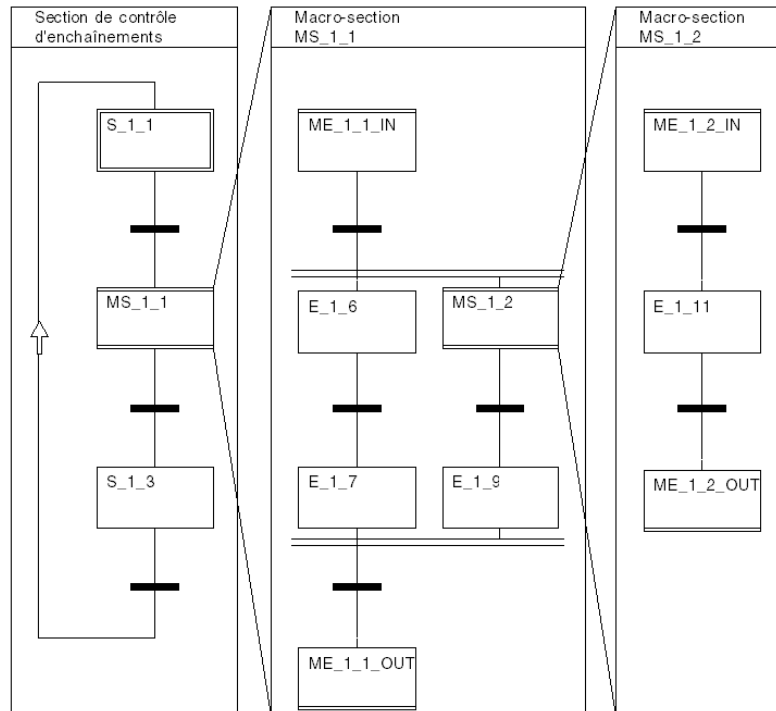
De plus, chaque macro-section comporte en début une étape d'entrée et en fin une étape de sortie.

Chaque macroétape peut être remplacée par la séquence contenue dans la macro-section correspondante.

Ainsi les macro-sections peuvent comporter 0, 1 ou plusieurs étapes initiales, voir également *Types d'étape*, page 406.

- Jeton unique
  - 0 étape(s) initiale(s)  
sont utilisées dans les macro-sections si une étape initiale est déjà disponible dans la section de rang hiérarchique supérieur ou inférieur.
  - 1 étape(s) initiale(s)  
sont utilisées dans les macro-sections si aucune étape initiale n'est disponible dans la section de rang hiérarchique supérieur ou inférieur.
- Jetons multiples  
Un maximum de 100 étapes initiales peuvent être utilisées dans chaque section (y compris toutes leurs macro-sections).

## Utilisation de macro-sections :



Le nom des macro-sections est identique au nom de la macroétape appelante. Si le nom de la macroétape est modifié, le nom de la macro-section correspondante l'est aussi automatiquement.

Une macro-section ne peut être utilisée qu'une seule fois.

## Exécution de macroétapes

Exécution des macroétapes :

Phase	Description
1	Une macroétape devient active lorsque la condition de transition qui la précède est VRAIE. L'étape d'entrée de la macro-section devient également active.
2	La séquence de la macro-section est exécutée. La macro-étape reste active tant qu'au moins une étape de la macro-section est active.
3	Si l'étape de sortie de la macro-section est active, la transition suivant la macroétape est validée.
4	La macroétape devient inactive lorsque l'étape de sortie est active et donc que la condition de transition suivante est validée et que la condition de transition est VRAIE. L'étape de sortie de la macro-section devient également inactive.

## Noms d'étape

A chaque création d'étape, un numéro de proposition lui est affecté.

Signification des numéros de proposition :

Type d'étape	Numéro de proposition	Description
Macroétape	ME_i_j	ME = Macroétape i = numéro courant (interne) de la section actuelle j = numéro de macro-étape courant (interne) de la section actuelle
Etape d'entrée	ME_k_l_IN	ME = Macroétape k = numéro courant (interne) de la section qui appelle l = numéro de macro-étape courant (interne) dans la section qui appelle IN = étape d'entrée
Etape de sortie	ME_k_l_OUT	ME = Macroétape k = numéro courant (interne) de la section qui appelle l = numéro de macro-étape courant (interne) dans la section qui appelle OUT = étape de sortie
Etape "normale" (dans une macro-section)	E_k_m	E = étape k = numéro courant (interne) de la section qui appelle m = numéro d'étape courant (interne) de la section appelante

Vous pouvez changer ces numéros de proposition pour avoir une meilleure vue d'ensemble. Les noms d'étape (28 caractères maximum pour les noms de macroétapes, 32 caractères maximum pour les noms d'étapes) doivent être uniques dans l'ensemble du projet, c.-à-d. qu'il ne doit pas exister d'autres étapes, variables, sections (excepté le nom de la macro-section affectée à la macroétape) etc., ayant le même nom. Aucune distinction n'est faite entre l'écriture en majuscules et en minuscules. Le nom d'étape doit répondre aux conventions sur les noms.

Si le nom de la macroétape est modifié, le nom de la macro-section correspondante et des étapes qu'elle contient le sont aussi automatiquement..

Exemple : si ME\_1\_1 est renommé en MyStep, les noms d'étape de la macro-section sont transformés en MyStep\_IN, MyStep\_1, ..., MyStep\_n, MyStep\_OUT.



---

## 13.3 Action et section d'action

---

### Objet de ce sous-chapitre

Ce sous-chapitre décrit les actions et les sections d'action du diagramme fonctionnel en séquence SFC

### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Action	416
Section d'action	418
Qualificatif	419

## Action

### Présentation

Les actions ont les caractéristiques suivantes :

- une action peut être une variable booléenne (variable d'action (*voir page 417*)) ou une section (section d'action (*voir page 418*)) du langage de programmation FBD, LD, IL ou ST.
- il est possible d'affecter aucune ou plusieurs actions à une étape. Une étape à laquelle aucune action n'est affectée a une fonction d'attente, c.-à-d. qu'elle attend que la transition affectée soit vérifiée.
- si plusieurs actions sont affectées à une même étape, elles sont traitées dans l'ordre de leur occurrence dans la zone de liste action.  
Exception : Indépendamment de leur position dans la zone de liste action, les actions comportant l'identificateur  $P1$  sont toujours traitées en premier et les actions comportant l'identificateur  $P0$  sont toujours traitées en dernier.
- la commande des actions est exprimée par des identificateurs (*voir page 419*).
- un maximum de 20 actions peuvent être affectées à chaque étape.
- la variable affectée à une action peut également être utilisée dans des actions d'autres étapes.
- vous pouvez utiliser la variable d'action en lecture et en écriture dans autant de sections du projet que vous désirez (affectation multiple !).
- les actions auxquelles un identificateur temporisé a été affecté ne peuvent être actives qu'une seule fois.
- seules les variables/adresses booléennes ou les éléments booléens de variables multi-éléments peuvent être utilisés comme variable d'action.
- les actions ont un nom unique.  
le nom de l'action est soit le nom de la variable d'action, soit le nom de la section d'action.

## Variable d'action

Sont autorisées comme variables d'action :

- Adresse du type de données `BOOL`  
Une action peut être affectée à une sortie matérielle via une adresse. Dans ce cas, l'action peut être utilisée en tant que signal de validation d'une transition, en tant que signal d'entrée dans une autre section ou en tant que signal de sortie pour le matériel.
- Variable simple ou élément d'une variable multi-élément du type de donnée `BOOL`  
Une variable permet d'utiliser une action comme signal d'entrée dans une autre section.
  - Variable non localisée  
Pour les variables non localisées, l'action peut être utilisée en tant que signal de validation d'une transition et en tant que signal d'entrée dans une autre section.
  - Variable localisée  
Pour les variables localisées, l'action peut être utilisée comme signal de validation d'une transition, comme signal d'entrée dans une autre section et comme signal de sortie pour le matériel.

## Noms d'action

Si vous utilisez une adresse ou une variable en tant qu'action, sa désignation sert de nom d'action (p.ex. %Q10.4, Variable1).

Si vous utilisez une section d'action en tant qu'action, le nom de section sert de nom d'action.

Les noms d'action (32 caractères maximum) doivent être uniques dans l'ensemble du projet, c.-à-d. qu'il ne doit pas exister d'autres transitions, variables, sections etc., ayant le même nom. Aucune distinction n'est faite entre l'écriture en majuscules et en minuscules. Le nom d'action doit répondre aux conventions sur les noms.

## Section d'action

### Présentation

Pour chaque action vous pouvez créer une section d'action. Il s'agit d'une section qui contient la logique de l'action et qui est automatiquement reliée à l'action.

### Nom de la section d'action

Le nom de la section d'action est toujours identique au nom de l'action correspondante, voir *Noms d'action*, page 417.

### Langages de programmation

Les langages de programmation possibles pour les sections d'action sont FBD, LD, IL et ST.

### Caractéristiques des sections d'action

Les sections d'action ont les caractéristiques suivantes :

- elles peuvent comporter un nombre quelconque de sorties.
- les appels de sous-programmes ne sont possibles dans les sections d'action que si le mode Multitoken a été activé.

**Remarque :** Les sous-programmes appelés ne sont **pas** soumis à la commande de la séquence, cela signifie que

- l'identificateur affecté à la section d'action appelante n'a aucun effet sur le sous-programme,
- le sous-programme reste actif, même à l'issue de la désactivation de l'étape appelante.
- les fonctions et blocs fonction ou procédures de diagnostic ne peuvent pas être utilisés dans les sections d'action.
- les sections d'action peuvent comporter de multiples réseaux.
- les sections d'action appartiennent à la section SFC dans laquelle elles ont été définies et peuvent dans cette section SFC (et toutes ses macro-sections) être affectées à un nombre quelconque d'actions.
- les sections d'action auxquelles un identificateur à durée a été affecté ne peuvent être actives qu'une seule fois.
- les sections d'action font partie de la section SFC dans laquelle elles ont été définies. Si la section SFC correspondante est supprimée, toutes les sections d'action de cette section SFC sont supprimées automatiquement.
- les sections d'action peuvent être appelées uniquement par des actions.

## Qualificatif

### Présentation

A chaque connexion d'une action à une étape, il faut définir pour l'action un identificateur définissant la commande de l'action.

### Identificateurs disponibles

Les identificateurs suivants sont disponibles :

Identificateur	Signification	Description
N / Aucun	Non mémorisé	Lorsque l'étape devient active, l'action passe à 1. Lorsque l'étape devient inactive, l'action tombe à 0.
R	Réinitialisation prioritaire	L'action définie dans une autre étape avec l'identificateur S est réinitialisée. En outre, il est possible d'éviter l'activation de n'importe quelle action. <b>Remarque :</b> les identificateurs sont déclarés automatiquement sans mémoire tampon. Cela signifie que leur valeur est remise à zéro après l'arrêt et le redémarrage du programme, p. ex. après une mise hors, puis sous tension. Si vous avez besoin d'une sortie avec une mémoire tampon, utilisez les blocs fonction RS ou SR de la bibliothèque de blocs standard.

Identificateur	Signification	Description				
S	Définition (méorisée)	<p>Une action définie reste active lorsque l'étape correspondante devient inactive. L'action ne devient inactive que lorsqu'elle est réinitialisée dans une autre étape de la séquence en cours avec l'identificateur R. Cela signifie, par exemple, que la réinitialisation de l'action à partir d'une autre section n'est pas possible.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Section_1</th> <th style="width: 50%;">Section_2</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;"> </td> <td style="vertical-align: top;"> <pre>IF S2.x= true THEN   A1:= false; END_IF;</pre> </td> </tr> </tbody> </table> </div> <div style="margin: 10px 0;"> </div> <p><b>Remarque :</b> les identificateurs sont déclarés automatiquement sans mémoire tampon. Cela signifie que leur valeur est remise à zéro après l'arrêt et le redémarrage du programme, p. ex. après une mise hors, puis sous tension. Si vous avez besoin d'une sortie avec une mémoire tampon, utilisez les blocs fonction RS ou SR de la bibliothèque de blocs standard.</p> <p><b>Remarque :</b> Il est possible d'attribuer jusqu'à 100 actions dont l'identificateur est S par section SFC.</p>	Section_1	Section_2		<pre>IF S2.x= true THEN   A1:= false; END_IF;</pre>
Section_1	Section_2					
	<pre>IF S2.x= true THEN   A1:= false; END_IF;</pre>					
L	limité dans le temps	<p>Lorsque l'étape devient active, l'action le devient également. Lorsque la durée que vous avez définie pour l'action est écoulée, l'action repasse à 0, même si l'étape est encore active. L'action passe également à 0 lorsque l'étape devient inactive.</p> <p><b>Remarque :</b> pour cet identificateur, il faut de plus définir une durée du type TIME.</p>				

Identificateur	Signification	Description
D	retardé	Lorsque l'étape devient active, le temporisateur interne est lancé et l'action est définie sur 1 après la durée définie manuellement pour l'action. Ensuite, lorsque l'étape devient inactive, l'action devient également inactive. Si l'étape devient inactive avant écoulement du temps interne, l'action ne devient pas active. <b>Remarque :</b> pour cet identificateur, il faut de plus définir une durée du type <code>TIME</code> .
P	Impulsion	Lorsque l'étape devient active, l'action passe à 1 et le reste pendant un cycle de programme, indépendamment du fait que l'étape reste active ou non.
DS	retardé et mémorisé	Lorsque l'étape devient active, le temporisateur interne est lancé et lorsque la durée définie est écoulée, l'action devient active. L'action ne redevient inactive que lorsqu'elle est réinitialisée dans une autre étape avec l'identificateur R. Si l'étape devient inactive avant écoulement du temps interne, l'action ne devient pas active. <b>Remarque :</b> pour cet identificateur, il faut de plus définir une durée du type <code>TIME</code> .
P1	Impulsion (front montant)	Lorsque l'étape devient active (front 0->1), l'action passe à 1 et le reste pendant un cycle de programme, indépendamment du fait que l'étape reste active ou non. <b>Remarque :</b> Indépendamment de leur position dans la zone de liste des actions, les actions avec l'identificateur <code>P1</code> sont toujours exécutées en premier, voir également <i>Action, page 416</i> .
P0	Impulsion (front descendant)	Lorsque l'étape devient inactive (front 1->0), l'action passe à 1 et le reste pendant un cycle de programme. <b>Remarque :</b> Indépendamment de leur position dans la zone de liste des actions, les actions avec l'identificateur <code>P0</code> sont toujours exécutées en dernier, voir également <i>Action, page 416</i> .

## 13.4 Transition et section de transition

---

### Objet de ce sous-chapitre

Ce sous-chapitre décrit les objets transition et les sections de transition du diagramme fonctionnel en séquence SFC.

### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Transition	423
Section de transition	425



## Transition

### Présentation

Une transition indique la condition à laquelle le contrôle d'une ou de plusieurs étapes précédant la transition passe à une ou plusieurs étapes subséquentes, le long de la liaison correspondante.

### Condition de transition

A chaque transition est affectée une condition de transition du type de donnée `BOOL`.

Sont autorisés comme conditions de transition :

- une adresse (entrée ou sortie),
- une variable (entrée ou sortie),
- un libellé ou
- une section de transition (*voir page 425*).

Le type de condition de transition détermine la position de son nom.

Condition de transition	Position du nom
<ul style="list-style-type: none"> <li>● Adresse</li> <li>● Variable</li> </ul>	
<ul style="list-style-type: none"> <li>● Libellé</li> </ul>	
<ul style="list-style-type: none"> <li>● Section de transition</li> </ul>	

### Noms de transition

Si vous utilisez une adresse ou une variable comme condition de transition, sa désignation est utilisée comme nom de transition (p. ex. `%I10.4, Variable1`).

Si vous utilisez une section de transition comme condition de transition, le nom de section est utilisé comme nom de transition.

Les noms de transition (32 caractères maximum) doivent être uniques dans l'ensemble du projet, c.-à-d. qu'il ne doit pas exister d'autres transitions, variables, sections etc. (à l'exception de la section de transition correspondante) ayant le même nom. Aucune distinction n'est faite entre l'écriture en majuscules et en minuscules. Le nom de transition doit répondre aux conventions sur les noms.

### **Validation d'une transition**

Une transition est validée lorsque les étapes qui la précèdent directement sont actives. Les transitions, dont les étapes qui les précèdent directement ne sont pas actives, ne sont normalement pas évaluées.

**NOTE** : Si aucune condition de transition n'est définie, la transition ne sera jamais active.

### **Déclenchement d'une transition**

Le déclenchement d'une transition s'effectue lorsque la transition est validée et que la condition de transition correspondante est vraie.

Le déclenchement d'une transition entraîne la désactivation (réinitialisation) de toutes les étapes qui la précèdent directement et qui sont connectées à la transition. Ensuite, toutes les étapes qui la suivent directement sont activées.

### **Temps de déclenchement d'une transition**

Théoriquement, le temps de déclenchement (temps de commutation) d'une transition peut être considéré comme étant le plus court possible, mais il ne peut jamais prendre la valeur zéro. Le temps de déclenchement d'une transition dure au moins le temps d'un cycle de programme.

## Section de transition

### Présentation

Une section de transition peut être créée pour chaque transition. Il s'agit d'une section qui contient la logique de la condition de transition et qui est automatiquement reliée à la transition.

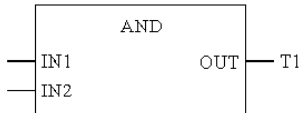

### Nom de la section de transition

Le nom de la section de transition est toujours identique au nom de la transition correspondante, voir *Noms de transition*, page 423.

### Langages de programmation

Les langages de programmation des sections de transition sont FBD , LD , IL et ST.

Réseaux conseillés pour les sections de transition :

Langage	Réseau conseillé	Description
FBD		<p>Le réseau conseillé contient un bloc ET comportant 2 entrées et dont la sortie booléenne est liée à une variable qui porte le nom de la section de transition.</p> <p>Le bloc proposé peut être relié ou, si vous ne le désirez pas, être supprimé.</p>
LD		<p>Le réseau proposé contient une bobine reliée à une variable portant le nom de la section de transition.</p> <p>La bobine proposée peut être reliée ou, si vous ne la désirez pas, être supprimée.</p>
IL	-	<p>Le réseau proposé est vide.</p> <p>Le seul contenu admis consiste en la création d'une logique booléenne. L'affectation du résultat logique à la sortie (variable de transition) est effectuée automatiquement, c-à-d. l'instruction mémoire ST n'est pas admise.</p> <p>Exemple :</p> <pre>LD A AND B</pre>

Langage	Réseau conseillé	Description
ST	-	<p>Le réseau proposé est vide.  Le seul contenu admis consiste à créer une logique booléenne sous forme d'expression (imbriquée). L'affectation du résultat logique à la sortie (variable de transition) est effectuée automatiquement, c-à-d. l'instruction d'affectation = n'est pas admise. L'expression ne se termine pas par un point-virgule (;).</p> <p>Exemple :</p> <pre>A AND B ou A AND (WORD_TO_BOOL (B))</pre>

### Caractéristiques des sections de transition

Les sections de transition ont les caractéristiques suivantes :

- les sections de transition n'ont qu'une seule sortie (la variable de transition) et son type de donnée est `BOOL`. Le nom de cette variable est identique au nom de la section de transition.
- la variable de transition ne doit être utilisée en écriture qu'une seule fois.
- la variable de transition peut être lue à n'importe quelle position du projet.
- seules des fonctions peuvent être utilisées et non pas des blocs fonction ni des procédures.
- dans LD, seule une bobine peut être utilisée.
- il n'existe qu'un seul réseau, c.-à-d. que toutes les fonctions utilisées sont reliées entre elles directement ou indirectement.
- les sections de transition ne peuvent être utilisées qu'une seule fois.
- les sections de transition font partie de la section SFC dans laquelle elles ont été définies. Si la section SFC correspondante est supprimée, toutes les sections de transition de ces sections SFC sont supprimées automatiquement.
- les sections de transition ne peuvent être appelées que par une transition.

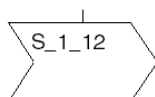
## 13.5 Saut

### Saut

#### Généralités

Les sauts permettent de représenter des liaisons dirigées qui ne sont pas tracées entièrement.

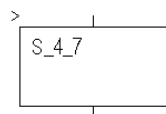
Représentation d'un saut :



#### Caractéristiques d'un saut

Un saut a les caractéristiques suivantes :

- plusieurs sauts peuvent avoir une même étape pour cible.
- selon la norme CEI 61131-3, les sauts dans une séquence en ET (*voir page 431*) ou depuis une séquence en ET ne sont pas autorisés. si vous voulez les utiliser tout de même, vous devez les valider explicitement.
- dans le cadre des sauts, on distingue les sauts de séquence (*voir page 436*) et les boucles de séquence (*voir page 437*).
- la cible du saut est munie du symbole du saut (>).



#### Nom de saut

Les sauts ne possèdent pas de nom en propre. Au lieu de cela, le nom de l'étape cible (cible du saut) est indiqué sur le symbole du saut.

## 13.6 Liaison

### Liaison

#### Présentation

Les liaisons relient les étapes et les transitions, les transitions et les étapes etc.

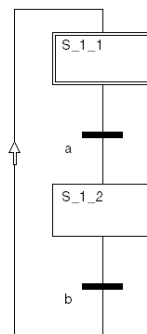
#### Caractéristiques des liaisons

Les liaisons ont les caractéristiques suivantes :

- il est impossible d'établir une liaison entre des objets de même type (étape avec étape, transition avec transition, etc.).
- une liaison peut être établie entre :
  - sorties d'objets non reliées et
  - entrées d'étape non reliées et reliées  
(cela signifie que une liaison multiple d'entrées d'étapes est possible)
- le chevauchement entre des liaisons et d'autres objets SFC (étape, transition, saut, etc.) n'est pas possible.
- les chevauchements entre liaisons sont possibles.
- Le croisement inter-liaison est possible et est indiqué sous forme de liaison "interrompue" :



- les liaisons se composent de segments verticaux et horizontaux.
- l'évolution de l'état logique dans une séquence est généralement de haut en bas. Cependant pour permettre la mise en oeuvre de boucles, les liaisons vers une étape peuvent également se faire de bas en haut. Ceci est valable pour les liaisons de transitions, de divergences en ET et de convergences en OU vers une étape. Dans ce cas, le sens de la liaison est représentée par une flèche.



- pour les liaisons, on distingue les sauts de séquence (*voir page 436*) et les boucles de séquence (*voir page 437*).

---

## 13.7 Divergences et convergences

---

### Objet de ce sous-chapitre

Ce sous-chapitre décrit les objets de convergence et de divergence du diagramme fonctionnel en séquence SFC.

### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Divergence en OU et convergence en OU	430
Divergence en ET et convergence en ET	431

## Divergence en OU et convergence en OU

### Présentation

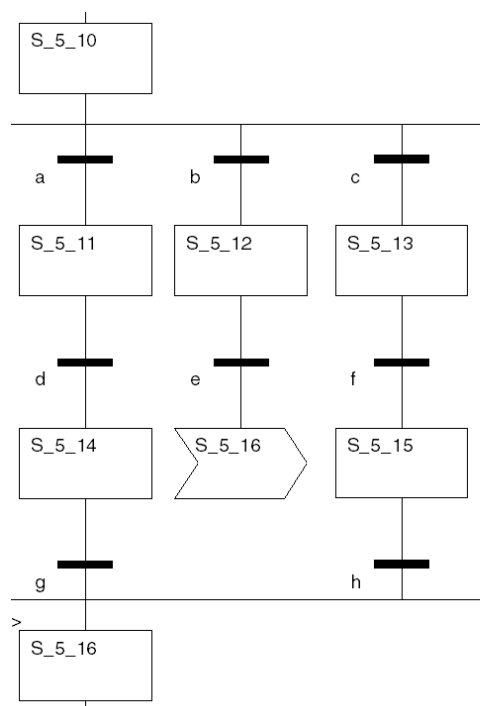
La divergence en OU permet de programmer des divergences à conditions au sein du flux de contrôle de la structure SFC.

Dans les divergences en OU, une étape est suivie d'autant de transitions sous la ligne horizontale qu'il y a d'enchaînements différents.

A l'aide de convergences en OU ou de sauts (*voir page 427*), toutes les branches alternatives convergent vers une branche unique dans laquelle le traitement se poursuit.

### Exemple d'une séquence en OU

Exemple d'une séquence en OU



### Caractéristiques d'une séquence en OU

Les caractéristiques d'une séquence en OU dépendent en grande partie du fait que la commande de séquence soit mise en oeuvre en jeton unique ou jetons multiples.

Voir

- Caractéristiques d'une séquence en OU en jeton unique (*voir page 435*)
- Caractéristiques d'une séquence en OU en jetons multiples (*voir page 447*)



## Divergence en ET et convergence en ET

### Présentation

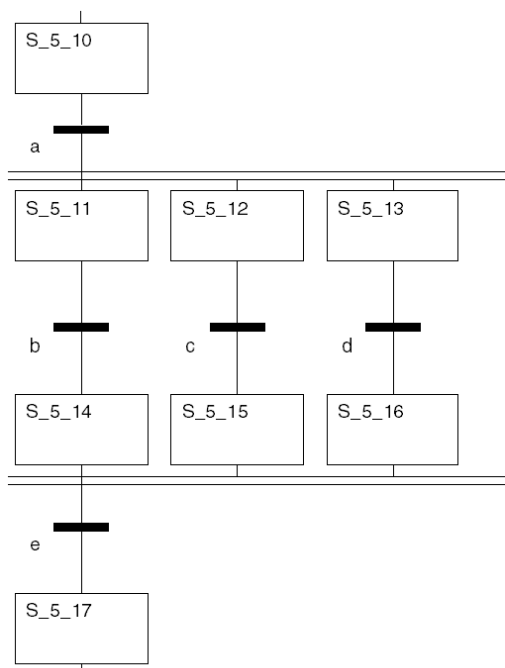
Avec les divergences en ET, la commutation d'une seule transition active plusieurs étapes (branches, 32 au maximum) en parallèle. Elles sont exécutées de gauche à droite. Après cette activation commune, les différentes branches sont exécutées indépendamment.

Conformément à la norme CEI 61131-1, toutes les branches d'une divergence en ET sont regroupées par une convergence en ET. La transition après une convergence en ET est évaluée lorsque toutes les étapes immédiatement précédentes de la convergence en ET ont été validées.

Si une divergence en ET doit être regroupée par une convergence en OU, ceci n'est possible qu'en mode jetons multiples (*voir page 450*).

### Exemple d'une séquence en ET

Exemple d'une séquence en ET



### Caractéristiques d'une séquence en ET

voir

- Caractéristiques d'une séquence en ET dans un jeton unique (*voir page 435*)
- Caractéristiques d'une séquence en ET dans un jeton multiple (*voir page 447*)

## **13.8           Objet texte**

---

### **Objet texte**

#### **Présentation**

En langage SFC, il est possible de positionner du texte sous forme d'objets texte. La taille de cet objet texte dépend de la longueur du texte. Cet objet texte a au moins la taille d'une cellule et peut, selon la taille du texte, être étendu sur des cellules complémentaires dans le sens vertical ou horizontal. Les objets de texte peuvent se recouvrir avec d'autres objets SFC.

---

## 13.9 Jeton unique

---

### Objet de ce sous-chapitre

Ce sous-chapitre décrit le mode "jeton unique" pour les commandes de séquence.

### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Jeton unique d'ordre d'exécution	434
Séquence en OU	435
Sauts de séquence et boucles de séquence	436
Séquences en ET	439
Sélection d'une séquence en ET asymétrique	441

## Jeton unique d'ordre d'exécution

### Description

Les règles suivantes s'appliquent aux jetons uniques :

- La situation d'origine est définie par l'étape initiale. La commande de séquence ne contient qu'une seule étape initiale.
- une seule étape est active à la fois dans une séquence. La seule exception est pour les divergences en ET dans lesquelles une étape est active par divergence.
- Les évolutions des états de signaux actifs se produisent le long des liaisons dirigées et sont déclenchés par la commutation d'une ou de plusieurs transitions. La séquence d'une chaîne va dans le sens des liaisons dirigées et se déroule de la partie inférieure de l'étape précédente à la partie supérieure de l'étape suivante.
- Une transition est validée lorsque les étapes qui la précèdent directement sont actives. Les transitions, dont les étapes qui précèdent directement ne sont pas actives, ne sont normalement pas évaluées.
- Le déclenchement d'une transition s'effectue lorsque la transition est validée et que la condition de transition correspondante est vraie.
- Le déclenchement d'une transition entraîne la désactivation (réinitialisation) de toutes les étapes qui la précèdent directement et qui sont connectées à la transition. Ensuite, toutes les étapes qui la suivent directement sont activées.
- Lorsque plusieurs conditions de transition d'une série d'étapes consécutives sont remplies, la scrutation est effectuée à une cadence d'une étape par cycle.
- Les étapes ne peuvent pas être activées ni désactivées par d'autres sections non SFC.
- Vous pouvez utiliser des macroétapes.
- Une seule branche est active à la fois dans une divergence en OU. Le résultat des conditions de transition détermine quelle branche est exécutée en fonction des transitions qui suivent la divergence en OU. Si une condition de transition est vraie, les transitions restantes ne sont plus exécutées. La branche qui contient la transition remplie devient active. Les divergences répondent à une priorité de gauche à droite. Toutes les divergences en OU sont regroupées à leur issue par une convergence en OU ou par des sauts.
- pour les divergences en ET, la commutation d'une seule transition active plusieurs étapes (branches) en parallèle. Après cette activation commune, les différentes branches sont exécutées indépendamment. Toutes les branches en ET sont regroupées à leur issue par une convergence en ET. Il n'est pas possible d'effectuer des sauts vers ou à partir d'une branche en ET.

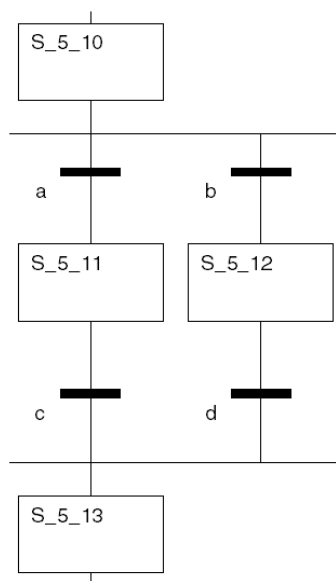
## Séquence en OU

### Séquence en OU

Selon la norme CEI 61131-3, seule une transition peut être vraie à la fois (1 parmi n choix). Le résultat des conditions de transition déterminera quelle branche sera exécutée en fonction des transitions qui suivent la divergence en OU. Les transitions de la divergence sont exécutées de gauche à droite. Si une condition de transition est vraie, les transitions restantes ne seront plus exécutées. La branche qui contient la transition remplie devient active. Il en résulte une priorité de gauche à droite pour les divergences.

Si aucune des transitions n'est vraie, l'étape validée à ce moment-là reste valide.

Séquence en OU :



Si...	Alors
Si S_5_10 est active et que la condition de transition a est vraie (indépendamment de b),	S_5_10 passe à S_5_11.
Si S_5_10 est active, que la condition de transition b est vraie et que a) est fausse,	S_5_10 passe à S_5_12.

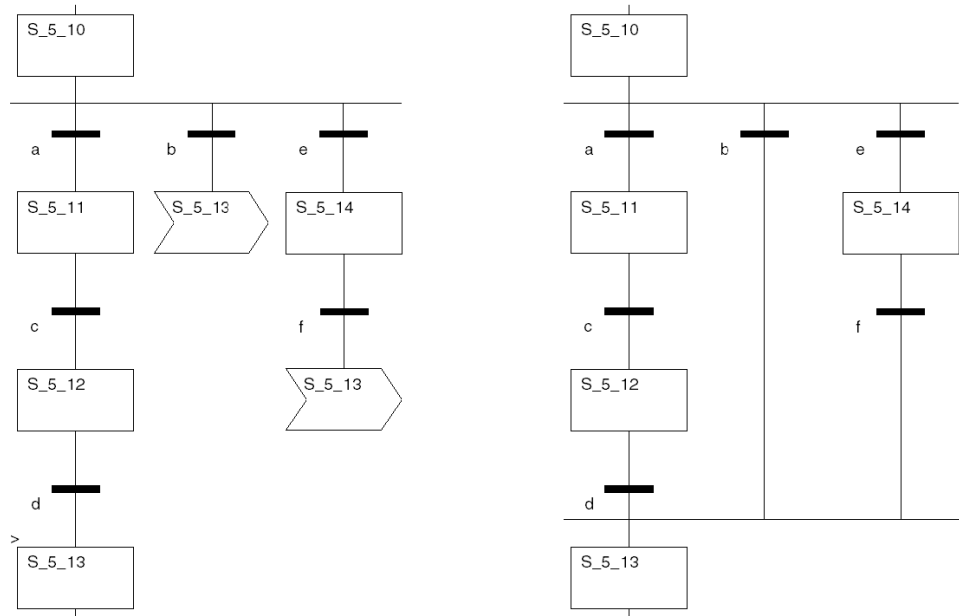
## Sauts de séquence et boucles de séquence

### Saut de séquence

Un saut de séquence est un cas particulier de la divergence en OU dans lequel quelques étapes de la séquence sont ignorées.

Un saut de séquence peut être réalisé avec des sauts ou avec des liaisons.

Saut de séquence :



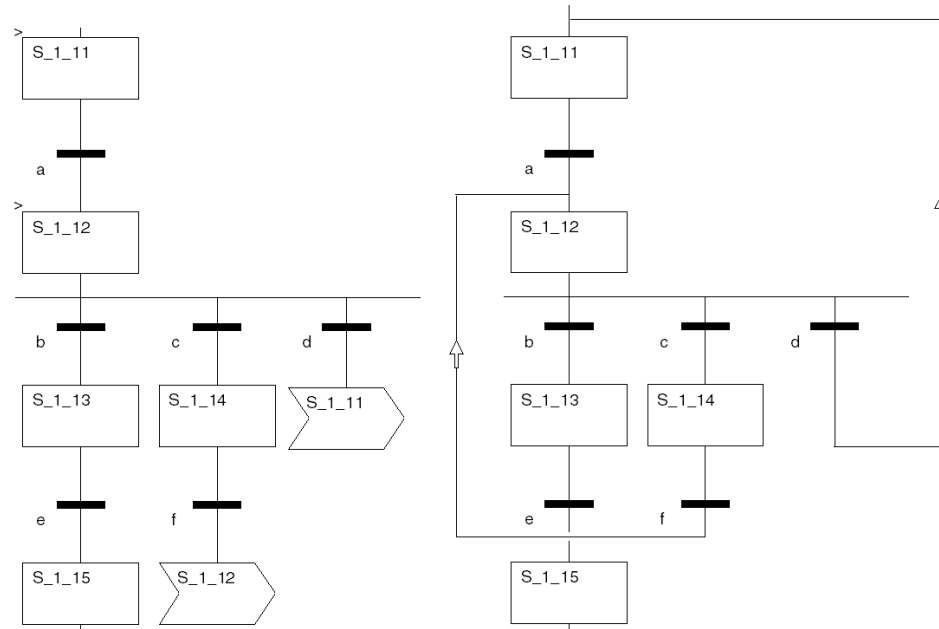
Si...	Alors
Si la condition de transition a est vraie,	la séquence passe de S_5_10 à S_5_11 puis S_5_12 et S_5_13.
Si la condition de transition b est vraie,	la séquence effectue un saut direct de S_5_10 vers S_5_13.
Si la condition de transition e est vraie,	la séquence passe de S_5_10 à S_5_14 et S_5_13.

## Boucle de séquence

Une séquence en boucle est un cas spécial de la dérivation en OU pour laquelle une ou plusieurs branches reconduisent à une étape précédente.

Une séquence en boucle peut être réalisée avec des sauts ou avec des liaisons.

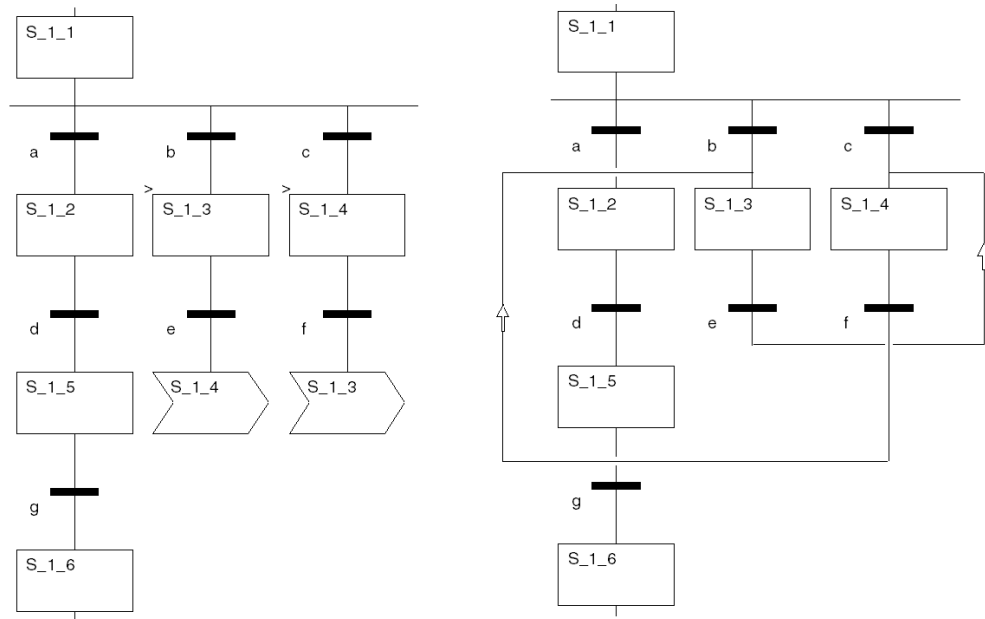
Boucle de séquence :



Si...	Alors
Si la condition de transition <i>a</i> est vraie,	la séquence passe de <i>S_1_11</i> à <i>S_1_12</i> .
Si la condition de transition <i>b</i> est vraie,	la séquence passe de <i>S_1_12</i> à <i>S_1_13</i> .
Si la condition de transition <i>b</i> est fausse et que <i>c</i> est vraie,	<i>S_1_12</i> est suivie de <i>S_1_14</i> .
Si la condition de transition <i>f</i> est vraie,	un saut est réalisé à partir de <i>S_1_14</i> jusque <i>S_1_12</i> .
La boucle de l'étape <i>S_1_12</i> via les conditions de transition <i>c</i> et <i>f</i> et le bouclage vers <i>S_1_12</i> est répétée jusqu'à ce que la condition de transition <i>b</i> soit vraie ou que <i>c</i> soit fausse et <i>d</i> vraie.	
Si les conditions de transition <i>b</i> et <i>c</i> sont fausses et que <i>d</i> est vraie,	<i>S_1_12</i> effectue un saut en arrière direct vers <i>S_1_11</i> .
La boucle de <i>S_1_11</i> vers <i>S_1_12</i> et retour vers <i>S_1_11</i> via les conditions de transition <i>a</i> et <i>d</i> est répétée jusqu'à ce que la condition de transition <i>b</i> ou <i>c</i> devienne vraie.	

Les séquences en boucle sans fin ne sont pas admises dans une séquence alternative.

Séquence en boucle sans fin :



Si...	Alors
Si la condition de transition b est vraie,	S_1_1 passe à S_1_3.
Si la condition de transition e est vraie,	un saut est exécuté vers S_1_4.
Si la condition de transition f est vraie,	un saut est exécuté vers S_1_3.
La boucle de S_1_3 via la condition de transition e vers S_1_4 via la condition de transition f et le saut de retour vers S_1_3 est alors répétée sans fin.	

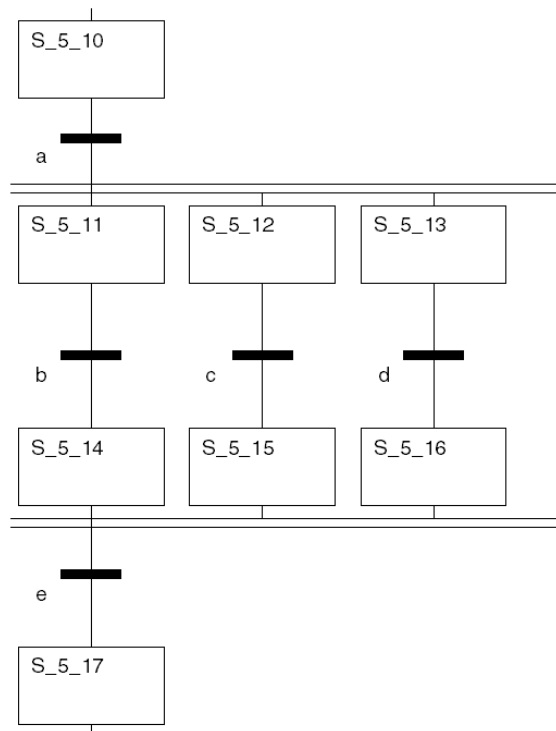


## Séquences en ET

### Séquences en ET

Pour les divergences en ET, la validation d'une seule transition active plusieurs étapes (branches, 32 au maximum) en parallèle. Ceci est valable aussi bien pour le jeton unique que pour les jetons multiples.

Exécution des séquences en ET :

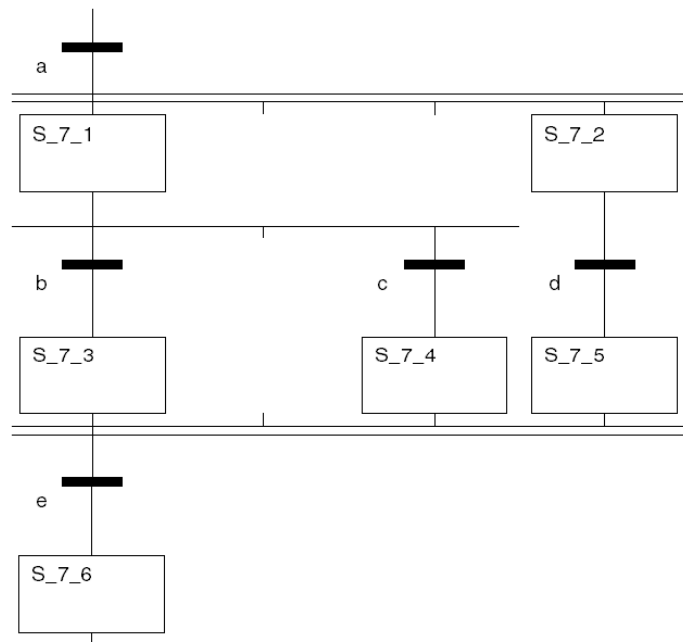


Si...	Alors
Si S_5_10 est active et que la condition de transition a, faisant partie de la transition commune, est également vraie,	S_5_10 passe à S_5_11, S_5_12 et S_5_13.
Si les étapes S_5_11, S_5_12 et S_5_13 sont activées,	les séquences sont exécutées indépendamment.
Si S_5_14, S_5_15 et S_5_16 sont actives simultanément et que la condition de transition e faisant partie de la transition commune est vraie,	S_5_14, S_5_15 et S_5_16 passent à S_5_17.

**Utilisation d'une divergence en OU dans une séquence en ET.**

Si vous utilisez une seule divergence en OU dans une séquence en ET, ceci provoque le blocage de la séquence dans le cas des jetons uniques.

Utilisation d'une divergence en OU dans une séquence en ET :



Si...	Alors
Si la condition de transition a est vraie,	la séquence passe à S_7_1 et S_7_2.
Si les étapes S_7_1 et S_7_2 sont activées,	les séquences sont exécutées indépendamment.
Si la condition de transition d est vraie,	S_7_5 est exécutée.
Si la condition de transition b est vraie et que c est fausse,	la séquence passe à S_7_3 est exécutée.
<p>S_7_3, S_7_4 et S_7_5 étant reliées par une convergence en ET, S_7_6 ne peut pas être exécutée puisque S_7_3 et S_7_4 ne peuvent jamais devenir actives en même temps. (Soit S_7_3 a été activée par la condition de transition b, soit S_7_4 a été activée par la condition de transition c, mais jamais les deux en même temps.)            C'est pourquoi S_7_3, S_7_4 et S_7_5 ne peuvent pas non plus devenir actives en même temps. La séquence se bloque.            Le même problème survient si lors de l'entrée dans la divergence en OU la condition de transition b est fausse et c est vraie.</p>	

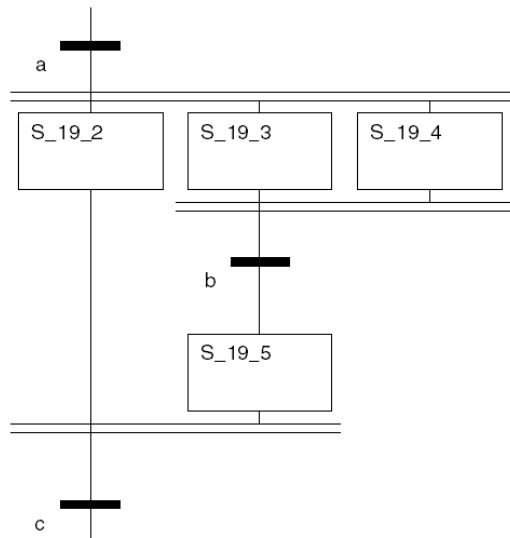
## Sélection d'une séquence en ET asymétrique

### Présentation

Selon la norme CEI 61131-3, une divergence en ET doit toujours être fermée par une convergence en ET. Le nombre de divergences en ET ne doit cependant pas nécessairement correspondre au nombre des convergences en ET.

### Nombre supérieur de convergences

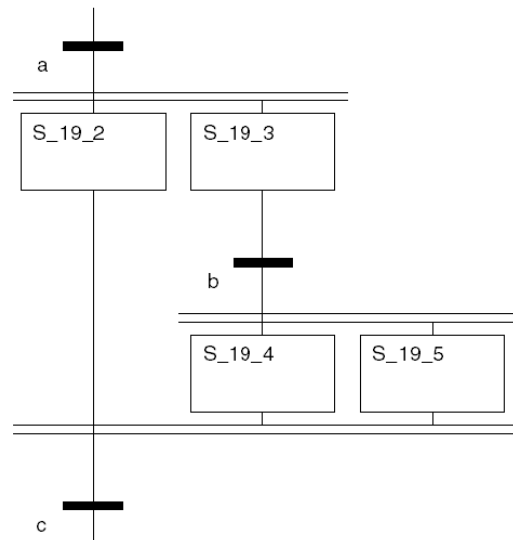
Séquence avec 1 divergence en ET et 2 convergences en ET



Si...	Alors
Si la condition de transition a est vraie,	S_19_2, S_19_3 et S_19_4 sont exécutées.
Si les étapes S_19_2, S_19_3 et S_19_4 sont activées,	les séquences sont exécutées indépendamment.
Si la condition de transition b est vraie,	S_19_5 est exécutée.
Si les étapes S_19_2 et S_19_5 sont actives et que la condition de transition c est vraie,	la séquence en ET est abandonnée.

**Nombre supérieur de divergences**

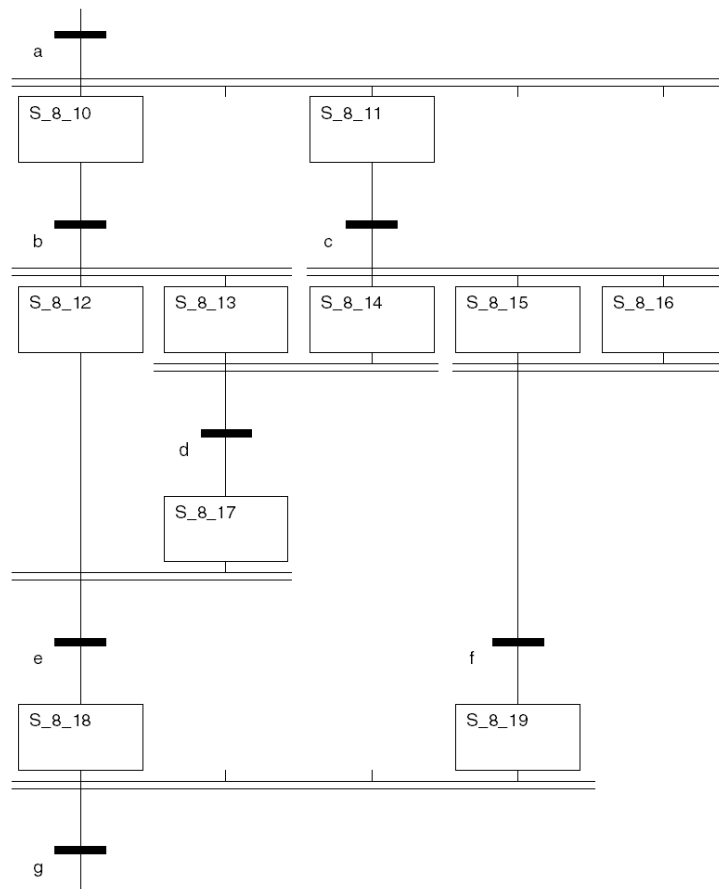
Séquence avec 2 divergences en ET et 1 convergence en ET



<b>Si...</b>	<b>Alors</b>
Si la condition de transition <b>a</b> est vraie,	<b>S_19_2</b> et <b>S_19_3</b> sont exécutées.
Si les étapes <b>S_19_2</b> et <b>S_19_3</b> sont activées,	les séquences sont exécutées indépendamment.
Si la condition de transition <b>b</b> est vraie,	<b>S_19_4</b> et <b>S_19_5</b> sont exécutées.
Si les étapes <b>S_19_4</b> et <b>S_19_5</b> sont activées,	les séquences sont exécutées indépendamment.
Si les étapes <b>S_19_2</b> , <b>S_19_4</b> et <b>S_19_5</b> sont actives et que la condition de transition <b>c</b> est vraie,	la séquence en ET est abandonnée.

## Séquences en ET imbriquées

Séquences en ET imbriquées :



Si...	Alors
Si la condition de transition a est vraie,	S_8_10 et S_8_11 sont exécutées.
Si la condition de transition b est vraie,	S_8_12 et S_8_13 sont exécutées.
Si la condition de transition c est vraie,	S_8_14, S_8_15 et S_8_16 sont exécutées.
Si les étapes S_8_13 et S_8_14 sont actives et que la condition de transition d est vraie,	S_8_17 est exécutée.
Si les étapes S_8_12 et S_8_17 sont actives et que la condition de transition e est vraie,	S_8_18 est exécutée.
...	...

## 13.10 Jetons multiples

---

### Objet de ce sous-chapitre

Ce sous-chapitre décrit le mode "jetons multiples" pour les commandes de séquence.

### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Ordre d'exécution à plusieurs jetons	445
Séquence en OU	447
Séquences en ET	450
Saut dans une séquence en ET	454
Saut hors d'une séquence en ET	455

## Ordre d'exécution à plusieurs jetons

### Description

Les règles suivantes s'appliquent aux jetons multiples :

- la situation de départ est définie par un nombre défini d'étapes initiales (de 0 à 100).
- dans la séquence, vous pouvez définir librement un nombre d'étapes qui seront simultanément actives.
- les évolutions des états de signaux actifs se produisent le long des liaisons dirigées et sont déclenchés par la commutation d'une ou de plusieurs transitions. La séquence d'une chaîne va dans le sens des liaisons dirigées et se déroule de la partie inférieure de l'étape précédente à la partie supérieure de l'étape suivante.
- une transition est validée lorsque les étapes qui la précèdent directement sont actives. Les transitions dont les étapes qui les précèdent directement ne sont pas actives, ne sont pas évaluées.
- le déclenchement d'une transition s'effectue lorsque la transition est validée et que la condition de transition correspondante est vraie.
- le déclenchement d'une transition entraîne la désactivation (réinitialisation) de toutes les étapes qui la précèdent directement et qui sont connectées à la transition. Ensuite, toutes les étapes qui la suivent directement sont activées.
- lorsque plusieurs conditions de transition d'une série d'étapes consécutives sont remplies, la scrutation est effectuée à une cadence d'une étape par cycle.
- les étapes et macro-étapes peuvent être activées ou désactivées par d'autres sections non SFC ou par des opérations utilisateur.
- si une étape active est simultanément activée et désactivée, l'étape reste active.
- vous pouvez utiliser des macro-étapes. Les sections de macro-étapes peuvent également comporter des étapes initiales.
- dans les divergences en OU, plusieurs branches peuvent être actives simultanément. Le résultat des conditions de transition des transitions qui suivent la divergence en OU déterminera quelle branche sera exécutée. Les transitions des branches sont exécutées en parallèle. Les branches dont la transition est remplie deviennent actives. Les branches en OU ne doivent pas être regroupées à leur issue par une convergence en OU ou par des sauts.
- si vous voulez mettre en oeuvre des sauts vers ou depuis une branche en ET, vous pouvez valider une option correspondante. Dans ce cas, les branches en ET ne doivent pas être regroupées à leur issue par une convergence en ET.
- des appels de sous-programmes peuvent être utilisés dans une section d'action.

- vous pouvez générer des jetons multiples à l'aide des éléments suivants :
  - étapes initiales multiples,
  - divergences en OU ou en ET non refermées,
  - sauts associés à des séquences en OU ou en ET,
  - activation d'étapes à l'aide du bloc de commande SFC `SETSTEP` depuis une section non-SFC ou à l'aide d'instructions de commande SFC.
- les jetons sont terminés par les actions suivantes :
  - rencontre simultanée de deux jetons ou plus dans une même étape,
  - désactivation d'étapes à l'aide du bloc de commande SFC `RESETSTEP` depuis une section non-SFC ou à l'aide d'instructions de commande SFC.



## Séquence en OU

### Séquence en OU

Dans le cas des jetons multiples, l'utilisateur peut définir le comportement d'évaluation des conditions de transitions dans les divergences en OU.

Les paramétrages possibles sont les suivants :

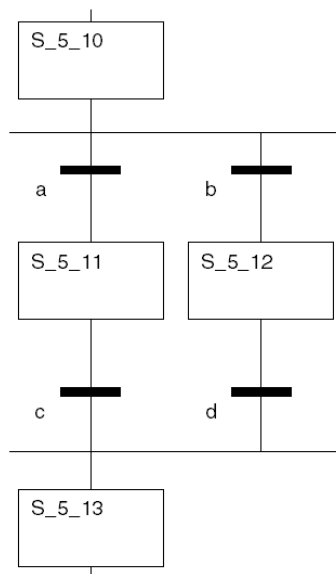
- Exécution de gauche à droite avec arrêt après la première transition active (choix de 1 parmi n). Ceci correspond au comportement des séquences en OU pour les Jetons uniques (*voir page 435*).
- Exécution en parallèle de toutes les transitions de la divergence en OU (choix de x parmi n)

### Choix de x parmi n

Pour le cas des jetons multiples, plusieurs transitions peuvent être validées parallèlement (choix de x parmi n). Le résultat des conditions de transition déterminera quelle branche sera exécutée en fonction des transitions qui suivent la divergence en OU. Les transitions des branches sont toutes exécutées. Toutes les branches dont la transition est remplie deviennent actives.

Si aucune des transitions n'est vraie, l'étape validée à ce moment-là reste valide.

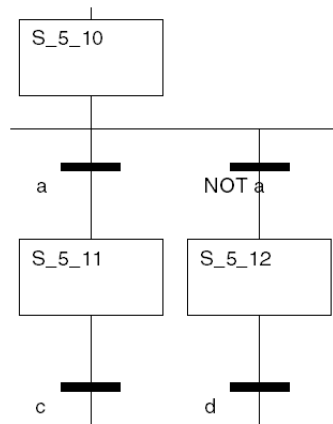
Choix de x parmi n :



Si...		Alors	
Si S_5_10 est active et que la condition de transition a est vraie et que b est fausse,		S_5_10 passe à S_5_11.	
Si S_5_10 est active et que la condition de transition a est fausse et b est vraie,		S_5_10 passe à S_5_12.	
Si S_5_10 est active et que les conditions de transition a et b sont vraies,		S_5_10 passe à S_5_11 et à S_5_12.	
L'activation en parallèle des deux branches en OU génère un deuxième jeton. Les deux jetons fonctionnent maintenant en parallèle, c.-à-d. S_5_11 et S_5_12 sont actives simultanément.			
Jeton 1 (S_5_11)		Jeton 2 (S_5_12)	
Si...	Alors	Si...	Alors
Si la condition de transition c est vraie,	S_5_11 passe à S_5_13.	Si la condition de transition d est vraie,	S_5_12 passe à S_5_13.
Si S_5_13 est encore active en raison de l'activation de la condition de transition c (jeton 1), le jeton 2 s'arrête et la séquence est exécutée à nouveau en jeton unique. Si S_5_13 n'est plus active (jeton 1), elle est réactivée par le jeton 2 et les deux jetons continuent à être exécutés en parallèle (jetons multiples).			

Si vous désirez que les divergences en OU ne soient validées que de manière exclusive dans ce mode, vous devez le définir explicitement dans la logique de transition.

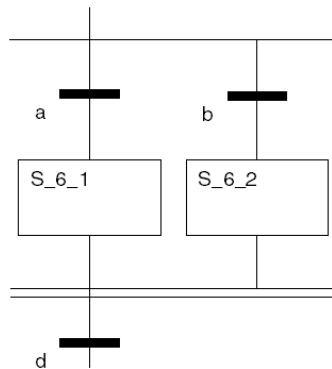
Exemple :



### Terminaison d'une divergence en OU par une convergence en ET

Si une divergence en OU est refermée par une convergence en ET, ceci peut déclencher un blocage de la séquence.

Terminaison d'une divergence en OU par une convergence en ET :



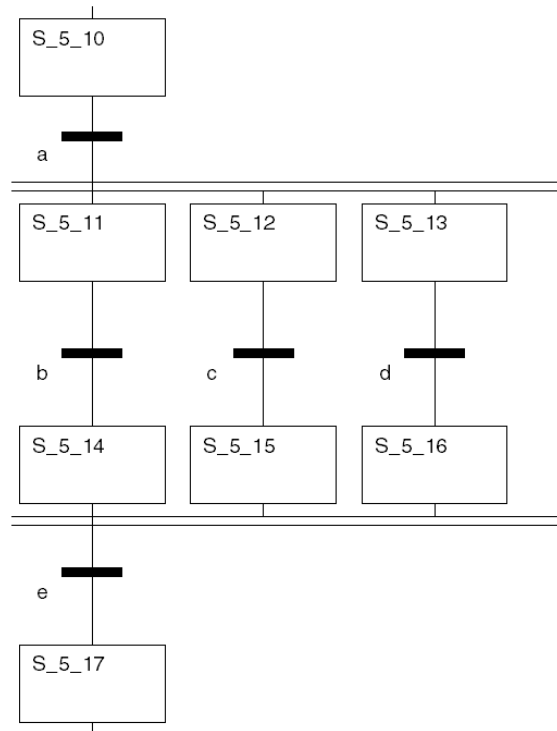
Si...	Alors
Si la condition de transition a est vraie et que b est fausse,	la séquence passe à S_6_1.
<p>S_6_1 et S_6_2 étant reliées par une convergence en ET, il n'est pas possible de quitter la divergence parce que S_6_1 et S_6_2 ne peuvent jamais devenir actives en même temps. (Soit S_6_1 a été activée par la condition de transition a soit S_6_2 a été activée par la condition de transition b.)</p> <p>C'est pourquoi S_6_1 et S_6_2 ne peuvent pas non plus devenir actives en même temps. La séquence se bloque.</p> <p>Ce blocage peut être levé p.ex. par un deuxième jeton décalé dans le temps, exécuté via la transition b.</p>	

## Séquences en ET

### Séquences en ET

Pour les divergences en ET, la validation d'une seule transition active plusieurs étapes (branches, 32 au maximum) en parallèle. Ceci est valable aussi bien pour le jeton unique que pour les jetons multiples.

Exécution des séquences en ET :

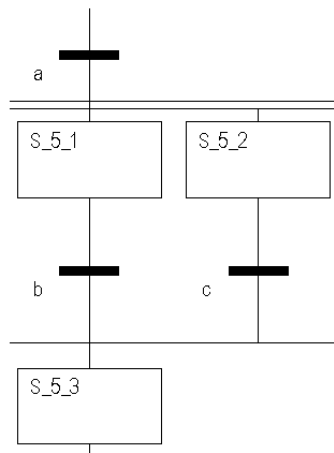


Si...	Alors
Si S_5_10 est active et que la condition de transition a, faisant partie de la transition commune, est également vraie,	S_5_10 passe à S_5_11, S_5_12 et S_5_13.
Si les étapes S_5_11, S_5_12 et S_5_13 sont activées,	les séquences sont exécutées indépendamment.
Si S_5_14, S_5_15 et S_5_16 sont actives simultanément et que la condition de transition e faisant partie de la transition commune est vraie,	S_5_14, S_5_15 et S_5_16 passent à S_5_17.

### Terminaison d'une divergence en ET par une convergence en OU

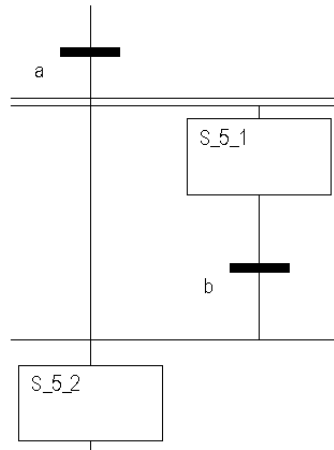
Pour les jetons multiples, il est possible de terminer les divergences en ET par une convergence en OU au lieu d'une convergence en ET.

Terminaison d'une séquence en ET par une divergence en OU (variante 1)



Si...		Alors	
Si la condition de transition a est vraie,		S_5_1 et S_5_2 sont exécutées.	
Si les étapes S_5_1 et S_5_2 sont activées,		les séquences sont exécutées indépendamment.	
Si la condition de transition b est vraie et que c est fausse,		S_5_3 est exécutée.	
L'exécution depuis la séquence en ET via la divergence en OU génère un deuxième jeton. Les deux jetons fonctionnent maintenant en parallèle, c.-à-d. S_5_2 et S_5_3 sont actives simultanément.			
Jeton 1 (S_5_3)		Jeton 2 (S_5_2)	
Si...	Alors	Si...	Alors
L'étape S_5_3 est active.		L'étape S_5_2 est active.	
		Si la condition de transition c est vraie,	S_5_3 est exécutée.
Si S_5_3 est encore active (jeton 1), le jeton 2 s'arrête et la séquence est exécutée à nouveau en jeton unique. Si S_5_3 n'est plus active (jeton 1), elle est réactivée par le jeton 2 et les deux jetons continuent à être exécutés en parallèle (jetons multiples).			

## Terminaison d'une séquence en ET par une divergence en OU (variante 2)

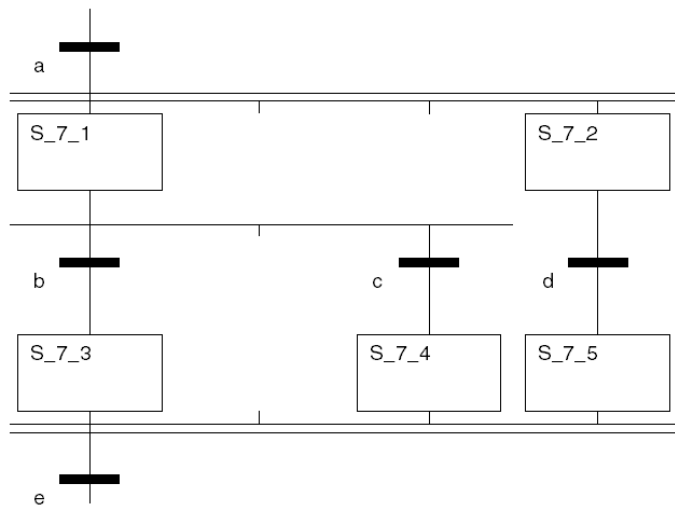


Si...		Alors	
Si la condition de transition a est vraie,		S_5_1 et S_5_2 sont exécutées.	
L'exécution depuis la séquence en ET via la divergence en OU génère un deuxième jeton. Les deux jetons fonctionnent maintenant en parallèle, c.-à-d. S_5_1 et S_5_2 sont actives simultanément.			
Jeton 1 (S_5_2)		Jeton 2 (S_5_1)	
Si...	Alors	Si...	Alors
L'étape S_5_2 est active.		L'étape S_5_1 est active.	
		Si la condition de transition b est vraie,	S_5_2 est exécutée.
Si S_5_2 est encore active (jeton 1), le jeton 2 s'arrête et la séquence est exécutée à nouveau en jeton unique. Si S_5_2 n'est plus active (jeton 1), elle est réactivée par le jeton 2 et les deux jetons continuent à être exécutés en parallèle (jetons multiples).			

### Utilisation d'une divergence en OU dans une séquence en ET.

Si une divergence en OU est terminée par une séquence en ET, ceci peut déclencher un blocage de la séquence.

Utilisation d'une divergence en OU dans une séquence en ET :



Si...	Alors
Si la condition de transition a est vraie,	la séquence passe à S_7_1 et S_7_2.
Si les étapes S_7_1 et S_7_2 sont activées,	les séquences sont exécutées indépendamment.
Si la condition de transition d est vraie,	S_7_5 est exécutée.
Si la condition de transition b est vraie,	la séquence passe à S_7_3.
<p>S_7_3, S_7_4 et S_7_5 étant reliées par une convergence en ET, il n'est pas possible de quitter la séquence en ET car S_7_3 et S_7_4 ne peuvent pas devenir actives en même temps.            (Soit S_7_3 a été activée par la condition de transition b, soit S_7_4 a été activée par la condition de transition c.)            C'est pourquoi S_7_3, S_7_4 et S_7_5 ne peuvent pas non plus devenir actives en même temps. La séquence se bloque.            Ce blocage peut être levé p.ex. par un deuxième jeton décalé dans le temps, exécuté via la transition c.</p>	

## Saut dans une séquence en ET

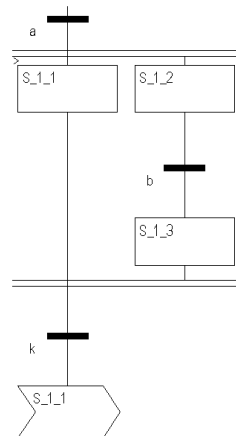
### Description

Pour les jetons multiples, vous pouvez valider au choix le saut dans une séquence en ET ou hors d'une séquence en ET.

Un saut dans une séquence en ET n'active pas toutes les branches. Suite à la convergence en ET la transition n'étant évaluée que lorsque toutes les étapes qui la précèdent directement ont été validées, vous ne pouvez plus quitter la séquence en ET, elle reste bloquée.

### Saut dans une séquence en ET

Saut dans une séquence en ET



Si...	Alors
Si la condition de transition a est vraie,	S_1_1 et S_1_2 sont exécutées.
Si les étapes S_1_1 et S_1_2 ont été activées,	les séquences sont exécutées indépendamment.
Si S_1_2 est active et que la condition de transition b est vraie,	la séquence passe de S_1_2 à S_1_3.
Si S_1_1 et S_1_3 sont actives et que la condition de transition c, de la transition commune est vraie,	S_1_1 et S_1_3 sont exécutées puis sautent vers S_1_1.
Si S_1_1 peut être activée par le saut,	seule la branche de S_1_1 devient active. La branche de S_1_2 ne devient pas active.
S_1_1 et S_1_3 ne devenant pas actives en même temps, la séquence ne peut pas continuer. Elle reste bloquée. Ce blocage peut être levé p. ex. par un deuxième jeton décalé dans le temps qui réactive l'étape S_1_2.	



## Saut hors d'une séquence en ET

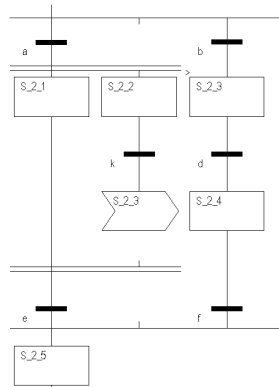
### Présentation

Pour les jetons multiples, vous pouvez valider au choix le saut dans une séquence en ET ou hors d'une séquence en ET.

Dans tous les cas, des jetons supplémentaires sont générés.

### Saut hors d'une séquence en ET.

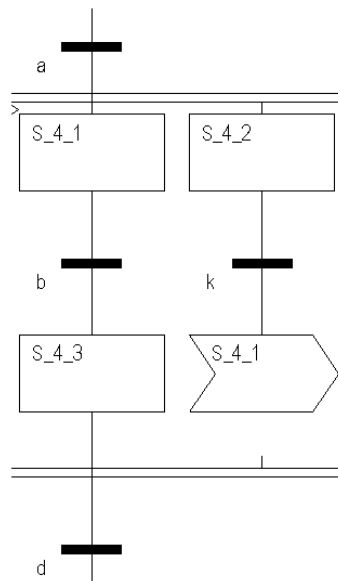
Saut hors d'une séquence en ET :



Si...		Alors	
Si la condition de transition a est vraie et que b est fausse,		S_2_1 et S_2_2 sont exécutées.	
Si les étapes S_2_1 et S_2_2 ont été activées,		les séquences sont exécutées indépendamment.	
Si la condition de transition c est vraie,		un saut est exécuté vers S_2_3.	
Le saut depuis la séquence en ET génère un deuxième jeton. Les deux jetons opèrent maintenant en parallèle, c.-à-d. S_2_1 et S_2_3 sont actives simultanément.			
Jeton 1 (S_2_1)		Jeton 2 (S_2_3)	
Si...	Alors	Si...	Alors
Si la condition de transition e est vraie,	la séquence passe à S_2_5.	Si la condition de transition d est vraie,	la séquence passe à S_2_4.
		Si la condition de transition f est vraie,	la séquence passe à S_2_5.
Si S_2_5 est encore active en raison de l'activation de la condition de transition e (jeton 1), le jeton 2 s'arrête et la séquence est exécutée à nouveau en jeton unique. Si S_2_5 n'est plus active (jeton 1), elle est réactivée par le jeton 2 et les deux jetons continuent à être exécutés en parallèle (jetons multiples).			

**Saut entre deux branches d'une séquence en ET.**

Saut entre deux branches d'une séquence en ET.



Si...		Alors	
Si la condition de transition a est vraie,		la séquence passe par S_4_1 et S_4_2.	
Si les étapes S_4_1 et S_4_2 sont activées,		les séquences sont exécutées indépendamment.	
Si la condition de transition b est vraie,		S_4_3 est exécutée.	
Si la condition de transition c est vraie,		un saut est exécuté vers S_4_1.	
Le saut depuis une branche d'une séquence en ET génère un deuxième jeton. Les deux jetons opèrent maintenant en parallèle, c.à-d. S_4_3 et S_4_1 sont maintenant actives en même temps.			
Jeton 1 (S_4_3)		Jeton 2 (S_4_1)	
Si...	Alors	Si...	Alors
l'étape S_4_3 est exécutée		l'étape S_4_1 est exécutée	
		Si la condition de transition b est vraie,	S_4_3 est exécutée.
Si lors de l'activation par le jeton 2, l'étape S_4_3 est encore active (jeton 1), le jeton 2 s'arrête et la séquence est exécutée à nouveau en jeton unique.			
Si lors de l'activation par le jeton 2, l'étape S_4_3 n'est plus active (jeton 1), elle est réactivée par le jeton 2 et les deux jetons continuent à être exécutés en parallèle (jetons multiples).			
Dans les deux cas, vous quittez la séquence en ET si la condition de transition d est vraie.			